Titulació:

Màster Universitari en Enginyeria Industrial (MUEI)

Alumne:

Guillem Llop Alonso

Estudi d'un sistema de supervisió remota basat en sensors intel·ligents

Director:

Miguel Delgado Prieto

Convocatòria de lliurament del TFG:

Ordinària; Quadrimestre de primavera 2018-2019

Data d'entrega:

20 de Juny del 2019

Contingut:

ANNEXES

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Taula de continguts

Annex I: Codi del programa

```
// Definició de variables principals. Posades al principi per a una fàcil edició

const short L = 2000;              // Nombre de mesures de x, y, z i C. Dimensió de les arrays
const short delayF = 500;          // Temps entre mesures en microsegons. Comporta una freqüència de 2 kHz
const short RPMtime = 4000;        // Temps durant el qual es mesuraran les rpm: 4 segons
const short cOffset = 0;           // Offset per corregir el soroll en la mesura de C
const short freq = 60000;          // Temps entre recollida total de mesures: un minut

bool consigna = 0;                 // Consigna d'entrada que permet funcionar al sistema

const float th1 = 0.71;            // Llindars per a la severitat de la vibració segons la ISO 2372
const float th2 = 1.8;             // Estan em mm/s, es comparen amb la RMS
const float th3 = 4.5;

String serie = "tfm2019";          // Nom del measurement enviat a InfluxDB

unsigned long t1 = 0;              // Timers utilitzats per assegurar la freqüència de mesura
unsigned long t2 = 0;              // Es comparen amb delayF

unsigned long timer = 0;           // Timers utilitzats per assegurar la periodicitat de mesures
unsigned long timer2 = 0;          // Es comparen amb freq

const int tachometerPin = A1;      // Entrada analògica del tacòmetre
const int cInput = A2;             // Entrada analògica del sensor de corrent
const int temperaturePin = 3;      // Entrada digital de la sonda de temperatura al motor
const int dhtPin = 4;              // Entrada digital del sensor de condicions ambientals
const int consignaPin = 5;         // Entrada digital de la consigna

const int xInput = A3;             // Entrada analògica de l'acceleròmetre: eix X
const int yInput = A4;             // Entrada analògica de l'acceleròmetre: eix Y
const int zInput = A5;             // Entrada analògica de l'acceleròmetre: eix Z

short xRawMin = 403;               // Llindars de calibració per a l'acceleròmetre...
short xRawMax = 600;               // ...obtinguts experimentalment

short yRawMin = 402;
short yRawMax = 600;

short zRawMin = 422;
short zRawMax = 618;

// Definicions per a la comunicació WiFi amb InfluxDB

#include <WiFiNINA.h>              // Llibreria

WiFiClient client;
IPAddress server(147,83,83,106);   // IP del servidor

char ssid[] = "wifiguillem";
char pass[] = "[redacted]";
int status = WL_IDLE_STATUS;       // Estat inicial de la ràdio WiFi

  String host = "147.83.83.106";   // IP del servidor
  String user = "guillem";         // Usuari de la base de dades
```

```
  String pw = "[redacted]";        // Contrasenya
  String db   = "guillem_tfm";     // Nom de la base de dades
  int port=11223;                  // Port de comunicacions de la base de dades

// Definicions per a la sonda de temperatura

#include <OneWire.h>                          // Llibreries
#include <DallasTemperature.h>

#define ONE_WIRE_BUS temperaturePin           // Pin d'Arduino associat

OneWire oneWire(ONE_WIRE_BUS);                // Creació de la instància oneWire

DallasTemperature sensors(&oneWire);          // Passem la referència oneWire com a referència...
                                              //...al sensor de Dallas Temperature

// Definicions per al sensor de condicons ambientals

#include <DHT.h>                              //Llibreria

#define DHTPIN dhtPin                         // Cable de dades connectat al port
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);                     // Inicialització del sensor DHT

// Definicions per al tacòmetre

bool fallen = false;                          // Flag per a la detecció de flancs de pujada o baixada
unsigned long lastTime = 0;                   // Últim moment en què s'ha detectat un flanc de pujada
unsigned long thisTime = 0;                   // Temps actual
float RPM = 0;                                // Valor de RPM

short lowLevel = 190;                  // Llindar inferior per flancs de baixada. Trobat experimentalment
short highLevel = 200;                 // Llindar superior per flancs de pujada. Trobat experimentalment

int thisReading;                       // Lectura del CNY-70 per al càlcul de RPM

// Definicions per a les vibracions i la corrent

short arrayX[L];                       // arrays de dades
short arrayY[L];
short arrayZ[L];
short arrayC[L];

short av;                              // Mitjana, utilitzada per centrar
short old1, old2;                      // Variables secundàries per al pas de velocitats a acceleracions

float minX, maxX, avX, rmsX;           // Estadístics de l'acceleració
float minY, maxY, avY, rmsY;
float minZ, maxZ, avZ, rmsZ;

float minVX, maxVX, avVX, rmsVX;       // Estadístics de la velocitat
float minVY, maxVY, avVY, rmsVY;
float minVZ, maxVZ, avVZ, rmsVZ;

float minC, maxC, avC, rmsC;           // Estadístics de la intensitat
```

```
float testOv;           // Valor utilitzat en la comprovació de desbordaments, que fan saltar alertes

byte sevX;              // Severitats de la vibració, segons la ISO 2372
byte sevY;
byte sevZ;

void setup()
{
 Serial.begin(9600);                    // Inicialització del port serial

 pinMode(tachometerPin, INPUT);         // Pins d'entrada com a INPUTS
 pinMode(xInput, INPUT);
 pinMode(yInput, INPUT);
 pinMode(zInput, INPUT);
 pinMode(cInput, INPUT);
 pinMode(consignaPin, INPUT);

 sensors.begin();                       // Inici del procés de la sonda de temperatura

 dht.begin();                           // Inici del procés del sensor ambiental

 while (!Serial) {                      // Esperem que es connecti el port serial
 }

 // Comprovar el bon funcionament del mòdul WiFi de l'Arduino:

if (WiFi.status() == WL_NO_MODULE) {
  Serial.println("La comunicació amb el mòdul WiFi ha fallat!");
  // No continuar
  while (true);
 }

 // Comprovar que el firmware està actualitzat

String fv = WiFi.firmwareVersion();
 if (fv < "1.0.0") {
  Serial.println("Sisplau, actualitzi el firmware");
 }

 // Intentar connectar-se a la xarxa WiFi:

while (status != WL_CONNECTED) {
  Serial.print("Intentant connectar amb la WPA SSID: ");
  Serial.println(ssid);

// Connectar-se a la xarxa WPA/WPA2:

  status = WiFi.begin(ssid,1,pass);

  // Esperar 5 segons perquè es faci la connexió:

  delay(5000);
 }

 // Ja estem connectats:
```

```
  Serial.print("Està connectat a la xarxa: ");
  printCurrentNet();
  Serial.println();

  delay(10000);                    // Espera per què la presa de dades no comenci tan ràpid.
}

void loop()
{
  consigna = digitalRead(consignaPin);              //Lectura de la consigna
  while (!consigna){                                // El programa no s'executa sense consigna = true
    delay(10000);                                   // Ho comprova cada 10 segons
    consigna = digitalRead(consignaPin);
  }

  timer2 = millis();

  while (timer2 - timer < freq){            // Aquí ens assegurem de prendre mesures periòdicament...
    timer2 = millis();                      // ...segons el valor de freq
  }

  timer = millis();

  t1=micros();                              // Presa de mesures de l'acceleròmetre en X
  for (int j=0; j<L; j++){
    t2=t1;
    while (t2-t1<delayF){                   // Ens assegurem una freqüència de mostreig...
      t2=micros();                          // ...segons el període delayF
    }
    t1=micros();
    arrayX[j] = analogRead(xInput);         // Lectura
  }

  t1=micros();                              // Presa de mesures de l'acceleròmetre en Y
  for (short j=0; j<L; j++){
    t2=t1;
    while (t2-t1<delayF){                   // Ens assegurem una freqüència de mostreig...
      t2=micros();                          // ...segons el període delayF
    }
    t1=micros();
    arrayY[j] = analogRead(yInput);
  }

  t1=micros();                              // Presa de mesures de l'acceleròmetre en Z
  for (short j=0; j<L; j++){
    t2=t1;
    while (t2-t1<delayF){                   // Ens assegurem una freqüència de mostreig...
      t2=micros();                           // ...segons el període delayF
    }
    t1=micros();
    arrayZ[j] = analogRead(zInput);         // Lectura
  }

  t1=micros();                              // Presa de mesures del sensor de corrent (C)
  for (short j=0; j<L; j++){
    t2=t1;
```

```
    while (t2-t1<delayF){                // Ens assegurem una freqüència de mostreig...
      t2=micros();                       // ...segons el período delayF
    }
    t1=micros();
    arrayC[j] = analogRead(cInput);      // Lectura
  }

  RPM = 0;                               // Presa de mesures del tacòmetre
  t1 = millis();
  t2 = t1;
  while (t2-t1<RPMtime){                 // Assegurem que la mesura es pren durant RPMtime
    t2=millis();
    thisReading = analogRead(tachometerPin);          // Lectura
    if (thisReading > highLevel){                     // Si tenim un valor alt...
      if (fallen == true){                            // ...després d'un flanc de baixada...
        thisTime = micros();                          // ...tenim flanc de pujada!
        RPM = 60000000.0 / (thisTime - lastTime);     // Guardem la dada
        lastTime = thisTime;
        fallen = false;
      }
    }
    if (thisReading < lowLevel){                      // Si tenim un valor baix...
      if (fallen == false){                           // ...després d'un flanc de pujada...
        fallen = true;                                // ...tenim flanc de baixada!
      }
    }
  }

// Recollida de temperatures i humitats

  sensors.requestTemperatures();
  float Tmotor = sensors.getTempCByIndex(0);          // Lectura del sensor de temperatures

  float Hamb = dht.readHumidity();                    // Lectura de la humitat ambiental
  float Tamb = dht.readTemperature();                 // Lectura de la temperatura ambiental

// *PROCESSAT DE X*

  for (short j=0; j<L; j++){
    arrayX[j] = map(arrayX[j], xRawMin, xRawMax, -1000, 1000);        //Passem a mG
  }

  for (short j=0; j<L; j++){
    testOv =arrayX[j]*9.80665;
    if (testOv>32767||testOv<-32768){                         // Test per evitar overflows
      alerta();                                       // Si hi ha overflow, salta l'alerta i para el programa
    }
    arrayX[j] = arrayX[j]*9.80665;                    // Passem a mm/s2
  }

  av = avArray(arrayX);
  for (short j=0; j<L; j++){
    arrayX[j] = arrayX[j] - av;                       // Centrem suposant mitjana zero en vibració
  }

  maxX = maxArray(arrayX)/1000.0;                     // Registrem paràmetres d'acceleració en m/s2
```

```
minX = minArray(arrayX)/1000.0;
avX = avArray(arrayX)/1000.0;
rmsX = rmsArray(arrayX)/1000.0;

old2 = arrayX[0];
arrayX[0] = 0;

for (short j=1; j<L; j++){                              // Aquí passarem d'acceleracions a velocitats
  old1 = old2;
  old2 = arrayX[j];
  testOv = arrayX[j-1] + (delayF/10000.0) * (old1+old2)/2.0;
  if (testOv>32767||testOv<-32768){                              // Test per evitar overflows
    alerta2();                                          // Si hi ha overflow, salta l'alerta i para el programa
  }
  arrayX[j] = arrayX[j-1] + (delayF/10000.0) * (old1+old2)/2.0;          // Velocitat en...
}                                                                        // ...desenes de micra/s (10^-5 s)

av = avArray(arrayX);
for (short j=0; j<L; j++){
  arrayX[j] = arrayX[j] - av;                           //Centrem suposant mitjana zero en vibració
}

maxVX = maxArray(arrayX)/100.0;                   // Registrem paràmetres de velocitats en mm/s
minVX = minArray(arrayX)/100.0;
avVX = avArray(arrayX)/100.0;
rmsVX = rmsArray(arrayX)/100.0;

if (rmsVX < th1){                                 // Assignem severitat segons els llindars
  sevX=1;
}

if (th2 > rmsVX && rmsVX >= th1){
  sevX=2;
}

  if (th3 > rmsVX && rmsVX >= th2){
  sevX=3;
}

  if (rmsVX >= th3){
  sevX=4;
}

// *PROCESSAT DE Y*

for (short j=0; j<L; j++){
  arrayY[j] = map(arrayY[j], yRawMin, yRawMax, -1000, 1000);          //Passem a mG
}
for (short j=0; j<L; j++){
  testOv=arrayY[j]*9.80665;
  if (testOv>32767||testOv<-32768){                              // Test per evitar overflows
    alerta();                                     // Si hi ha overflow, salta l'alerta i para el programa
  }
  arrayY[j] = arrayY[j]*9.80665;                  // Passem a mm/s2
}
av = avArray(arrayY);
```

```
  for (short j=0; j<L; j++){
    arrayY[j] = arrayY[j] - av;                    // Centrem suposant mitjana zero en vibració
  }
  maxY = maxArray(arrayY)/1000.0;                  // Registrem paràmetres d'acceleració en m/s2
  minY = minArray(arrayY)/1000.0;
  avY = avArray(arrayY)/1000.0;
  rmsY = rmsArray(arrayY)/1000.0;

  old2 = arrayY[0];
  arrayY[0] = 0;

  old2 = maxY;
  for (short j=1; j<L; j++){                       // Aquí passarem d'acceleracions a velocitats
    old1 = old2;
    old2 = arrayY[j];
    testOv=arrayY[j-1] + (delayF/10000.0) * (old1+old2)/2.0;
    if (testOv>32767||testOv<-32768){                         // Test per evitar overflows
      alerta2();                                    // Si hi ha overflow, salta l'alerta i para el programa
    }
    arrayY[j] = arrayY[j-1] + (delayF/10000.0) * (old1+old2)/2.0;          // Velocitat en...
  }                                                                        // ...desenes de micra/s (10^-5 s)

  av = avArray(arrayY);
  for (short j=0; j<L; j++){
    arrayY[j] = arrayY[j] - av;                    //Centrem suposant mitjana zero en vibració
  }

  maxVY = maxArray(arrayY)/100.0;                  // Registrem paràmetres de velocitats en mm/s
  minVY = minArray(arrayY)/100.0;
  avVY = avArray(arrayY)/100.0;
  rmsVY = rmsArray(arrayY)/100.0;

  if (rmsVY < th1){                                // Assignem severitat segons els llindars
    sevY=1;
  }

  if (th2 > rmsVY && rmsVY >= th1){
    sevY=2;
  }

    if (th3 > rmsVY && rmsVY >= th2){
    sevY=3;
  }

    if (rmsVY >= th3){
    sevY=4;
  }

// *PROCESSAT DE Z*

  for (short j=0; j<L; j++){
    arrayZ[j] = map(arrayZ[j], zRawMin, zRawMax, -1000, 1000);             //Passem a mG
  }
  for (short j=0; j<L; j++){
    testOv=arrayZ[j]*9.80665;
    if (testOv>32767||testOv<-32768){                         // Test per evitar overflows
```

```
    alerta();                                   // Si hi ha overflow, salta l'alerta i para el programa
  }
  arrayZ[j] = arrayZ[j]*9.80665;   // Passem a mm/s2
 }
 av = avArray(arrayZ);
 for (short j=0; j<L; j++){
  arrayZ[j] = arrayZ[j] - av;                // Centrem suposant mitjana zero en vibració
 }
 maxZ = maxArray(arrayZ)/1000.0;            // Registrem paràmetres d'acceleració en m/s2
 minZ = minArray(arrayZ)/1000.0;
 avZ = avArray(arrayZ)/1000.0;
 rmsZ = rmsArray(arrayZ)/1000.0;

 old2 = arrayZ[0];
 arrayZ[0] = 0;

 for (short j=1; j<L; j++){                        // Aquí passarem d'acceleracions a velocitats
  old1 = old2;
  old2 = arrayZ[j];
  testOv=arrayZ[j-1] + (delayF/10000.0) * (old1+old2)/2.0;
  if (testOv>32767||testOv<-32768){                            // Test per evitar overflows
    alerta2();                                         // Si hi ha overflow, salta l'alerta i para el programa
  }
  arrayZ[j] = arrayZ[j-1] + (delayF/10000.0) * (old1+old2)/2.0;           // Velocitat en...
 }                                                                       // ...desenes de micra/s (10^-5 s)

 av = avArray(arrayZ);
 for (short j=0; j<L; j++){
  arrayZ[j] = arrayZ[j] - av;                      //Centrem suposant mitjana zero en vibració
 }

 maxVZ = maxArray(arrayZ)/100.0;                   // Registrem paràmetres de velocitats en mm/s
 minVZ = minArray(arrayZ)/100.0;
 avVZ = avArray(arrayZ)/100.0;
 rmsVZ = rmsArray(arrayZ)/100.0;

 if (rmsVZ < th1){                         // Assignem severitat segons els llindars
  sevZ=1;
 }

 if (th2 > rmsVZ && rmsVZ >= th1){
  sevZ=2;
 }

  if (th3 > rmsVZ && rmsVZ >= th2){
  sevZ=3;
 }

  if (rmsVZ >= th3){
  sevZ=4;
 }

// Processat de C

 for (short j=0; j<L; j++){
  arrayC[j] = arrayC[j]*6600/1023-cOffset;          // Passem a mA
```

```
}

maxC = maxArray(arrayC);                            //Intensitat en mA
minC = minArray(arrayC);
avC = avArray(arrayC);
rmsC = rmsArray(arrayC);

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Serial.println();                                   // Imprimirem els principals estadístics pel port serial...
                                                    // ...per a poder comprovar el bon funcionament

Serial.print("X | Acceleració [m/s2]");
Serial.print("  Max: "); Serial.print(maxX);
Serial.print("  Min: "); Serial.print(minX);
Serial.print("  Mitjana: "); Serial.print(avX);
Serial.print("  RMS: "); Serial.println(rmsX);


Serial.print("X | Velocitat   [mm/s]");
Serial.print("  Max: "); Serial.print(maxVX);
Serial.print("  Min: "); Serial.print(minVX);
Serial.print("  Mitjana: "); Serial.print(avVX);
Serial.print("  RMS: "); Serial.print(rmsVX);
Serial.print("  Severitat: "); Serial.println(sevX);

Serial.println();

Serial.print("Y | Acceleració [m/s2]");
Serial.print("  Max: "); Serial.print(maxY);
Serial.print("  Min: "); Serial.print(minY);
Serial.print("  Mitjana: "); Serial.print(avY);
Serial.print("  RMS: "); Serial.println(rmsY);

Serial.print("Y | Velocitat   [mm/s]");
Serial.print("  Max: "); Serial.print(maxVY);
Serial.print("  Min: "); Serial.print(minVY);
Serial.print("  Mitjana: "); Serial.print(avVY);
Serial.print("  RMS: "); Serial.print(rmsVY);
Serial.print("  Severitat: "); Serial.println(sevY);

Serial.println();

Serial.print("Z | Acceleració [m/s2]");
Serial.print("  Max: "); Serial.print(maxZ);
Serial.print("  Min: "); Serial.print(minZ);
Serial.print("  Mitjana: "); Serial.print(avZ);
Serial.print("  RMS: "); Serial.println(rmsZ);

Serial.print("Z | Velocitat   [mm/s]");
Serial.print("  Max: "); Serial.print(maxVZ);
Serial.print("  Min: "); Serial.print(minVZ);
Serial.print("  Mitjana: "); Serial.print(avVZ);
Serial.print("  RMS: "); Serial.print(rmsVZ);
Serial.print("  Severitat: "); Serial.println(sevZ);
```

```
  Serial.println();

  Serial.print("C | Intensitat  [mA]  ");
  Serial.print("  Max: "); Serial.print(maxC);
  Serial.print("  Min: "); Serial.print(minC);
  Serial.print("  Mitjana: "); Serial.print(avC);
  Serial.print("  RMS: "); Serial.println(rmsC);

  Serial.println();
  Serial.print("RPM: ");
  Serial.println(RPM);

  Serial.print("Temperatura motor: ");
  Serial.print(Tmotor);

  if (isnan(Hamb) || isnan(Tamb)){
    Serial.println("  Error de lectura ambient");        // Si hi ha un error en la lectura...
  }                                                       // ...del DHT 22
  else{
    Serial.print("  Temperatura ambient: ");
    Serial.print(Tamb);
    Serial.print("  Humitat ambient: ");
    Serial.println(Hamb);
  }

  if (client.connect(server, port)) {                     //Iniciem la connexió al servidor
    Serial.println();
    Serial.println("Connecting...");
    Serial.println();                                     // Creem el cos de les dades a enviar:

    String data1 = (serie + " sevX=" + sevX + ",sevY=" + sevY + ",sevZ=" + sevZ);
    String data2 = (",maxVX=" + String(maxVX,2) + ",minVX=" + String(minVX,2) + ",avVX=" + String(avVX,2) +
",rmsVX=" + String(rmsVX,2));
    String data3 = (",maxVY=" + String(maxVY,2) + ",minVY=" + String(minVY,2) + ",avVY=" + String(avVY,2) +
",rmsVY=" + String(rmsVY,2));
    String data4 = (",maxVZ=" + String(maxVZ,2) + ",minVZ=" + String(minVZ,2) + ",avVZ=" + String(avVZ,2) +
",rmsVZ=" + String(rmsVZ,2));
    String data5 = (",maxC=" + String(maxC,2) + ",minC=" + String(minC,2) + ",avC=" + String(avC,2) + ",rmsC=" +
String(rmsC,2));
    String data6 = (",RPM=" + String(RPM,2) + ",Tmotor=" + String(Tmotor,2) + ",Tamb=" + String(Tamb,2) +
",Hamb=" + String(Hamb,2));
    String data = (data1 + data2 + data3 + data4 + data5 + data6);

    client.print("POST /write?u=" + user + "&p=" + pw + "&db=" + db);          // Fem la query
    client.println(" HTTP/1.1");
    client.println("Host: " + host);                                           // IP del servidor
    client.println("Content-Type: application/x-www-form-urlencoded");     // Tipus de contingut a enviar
    client.print("Content-length: ");                                       // Longitud del cos de les dades
    client.println(data.length());
    client.println("Connection: close");                                    // Acabem la connexió
    client.println();
    client.println(data);                                                   // Enviem el cos de les dades

    Serial.println(data);                                 // Mostrem el cos de les dades pel port serial...
                                                          // ...per comprovar el bon funcionament
```

```
  } else {                                          // Si no s'ha pogut connectar amb el servidor:
    Serial.println("connection failed");
  }

  Serial.println("--------------------");           // Línia de separació  entre lectures al port serial
}

short maxArray (short array[]){             // Funció: valor màxim d'una array
  short result=-32000;
  short k;
  for(k=0; k<L ; k++){
    if(array[k]>result){
      result=array[k];
    }
  }
  return result;
}

short minArray (short array[]){             // Funció: valor mínim d'una array
  short result=32000;
  short k;
  for(k=0; k<L ; k++){
    if(array[k]<result){
      result=array[k];
    }
  }
  return result;
}

short avArray (short array[]){              // Funció: mitjana d'una array
  short result;
  int sum=0;
  short k;
  for(k=0; k<L ; k++){
    sum = sum + array[k];
  }
  result = sum/L;
  return result;
}

short rmsArray (short array[]){             // Funció: RMS d'una array
  short result;
  double sum=0;
  short k;
  for(k=0; k<L ; k++){
    sum = sum + array[k]*array[k];
  }
  result = sqrt(sum/L);
  return result;
}

void alerta(){
  Serial.println("ALERTA overflow en la taula d'acceleracions");          // Funció d'alerta
  while(true){}
}
```

```
void alerta2(){
  Serial.println("ALERTA overflow en la taula de velocitats");                    // Funció d'alerta
  while(true){}
}

void printCurrentNet() {                    // Funció per imprimir el SSID de la xarxa WiFi on estem connectats
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
}
```

Annex II: *Datasheet* de l'ADXL 335

ANALOG DEVICES

Small, Low Power, 3-Axis ±3 *g* Accelerometer

ADXL335

FEATURES

3-axis sensing
Small, low profile package
4 mm × 4 mm × 1.45 mm LFCSP
Low power : 350 μA (typical)
Single-supply operation: 1.8 V to 3.6 V
10,000 *g* shock survival
Excellent temperature stability
BW adjustment with a single capacitor per axis
RoHS/WEEE lead-free compliant

APPLICATIONS

Cost sensitive, low power, motion- and tilt-sensing applications
Mobile devices
Gaming systems
Disk drive protection
Image stabilization
Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 *g*. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_X, C_Y, and C_Z capacitors at the X_{OUT}, Y_{OUT}, and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

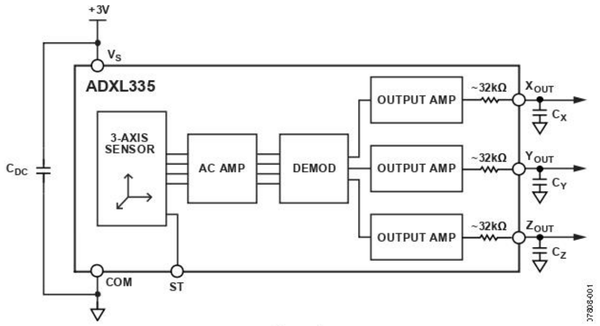


Figure 1.

Rev. 0

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

TABLE OF CONTENTS

REVISION HISTORY

1/09—Revision 0: Initial Version

SPECIFICATIONS

$T_A = 25°C$, $V_S = 3$ V, $C_X = C_Y = C_Z = 0.1$ µF, acceleration = 0 *g*, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		*g*
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[1]			±1		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/*g*
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	V_S = 3 V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		m*g*/°C
NOISE PERFORMANCE					
Noise Density X_{OUT}, Y_{OUT}			150		µ*g*/√Hz rms
Noise Density Z_{OUT}			300		µ*g*/√Hz rms
FREQUENCY RESPONSE[4]					
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter		1600		Hz
Bandwidth Z_{OUT}[5]	No external filter		550		Hz
R_{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST[6]					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	−150	−325	−600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V_S = 3 V		350		µA
Turn-On Time[7]	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		−40		+85	°C

[1] Defined as coupling between any two axes.
[2] Sensitivity is essentially ratiometric to V_S.
[3] Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.
[4] Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).
[5] Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 µF, bandwidth = 1.6 kHz. For C_Z = 0.01 µF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 µF, bandwidth = 0.5 Hz.
[6] Self-test response changes cubically with V_S.
[7] Turn-on time is dependent on C_X, C_Y, C_Z and is approximately 160 × C_X or C_Y or C_Z + 1 ms, where C_X, C_Y, C_Z are in microfarads (µF).

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 *g*
Acceleration (Any Axis, Powered)	10,000 *g*
V_S	−0.3 V to +3.6 V
All Other Pins	(COM − 0.3 V) to (V_S + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	−55°C to +125°C
Temperature Range (Storage)	−65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

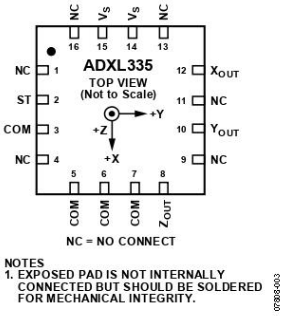


Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect[1].
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect[1].
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z_{OUT}	Z Channel Output.
9	NC	No Connect[1].
10	Y_{OUT}	Y Channel Output.
11	NC	No Connect[1].
12	X_{OUT}	X Channel Output.
13	NC	No Connect[1].
14	V_S	Supply Voltage (1.8 V to 3.6 V).
15	V_S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect[1].
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

[1]NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

ADXL335

TYPICAL PERFORMANCE CHARACTERISTICS

N > 1000 for all typical performance plots, unless otherwise noted.

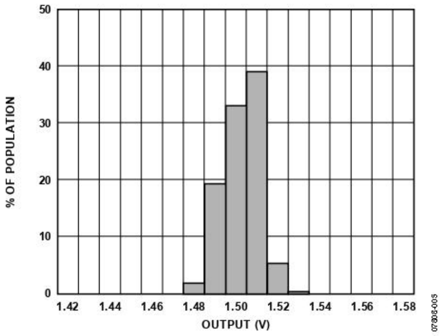


Figure 3. X-Axis Zero g Bias at 25°C, $V_S = 3$ V

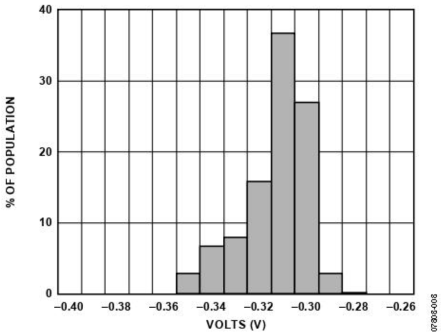


Figure 6. X-Axis Self-Test Response at 25°C, $V_S = 3$ V

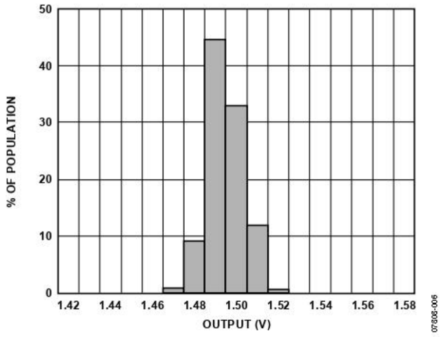


Figure 4. Y-Axis Zero g Bias at 25°C, $V_S = 3$ V

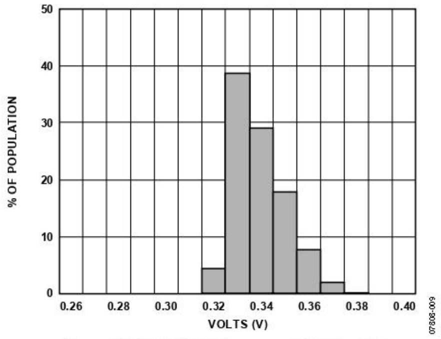


Figure 7. Y-Axis Self-Test Response at 25°C, $V_S = 3$ V

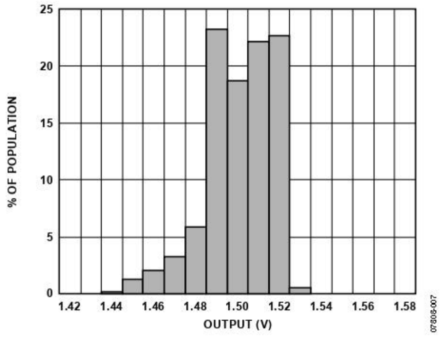


Figure 5. Z-Axis Zero g Bias at 25°C, $V_S = 3$ V

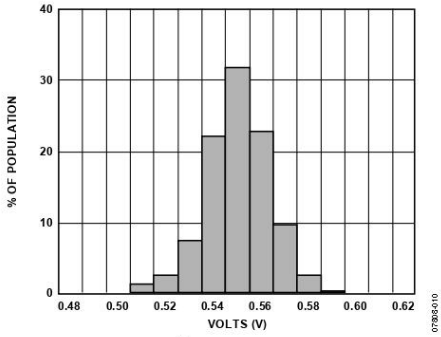


Figure 8. Z-Axis Self-Test Response at 25°C, $V_S = 3$ V

ADXL335

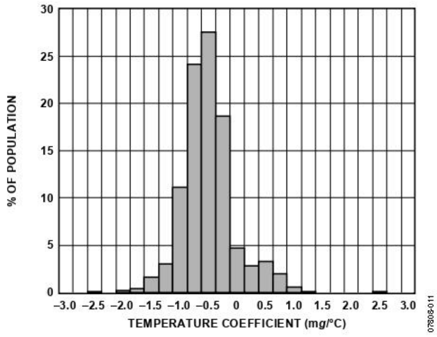


Figure 9. X-Axis Zero g Bias Temperature Coefficient, $V_S = 3$ V

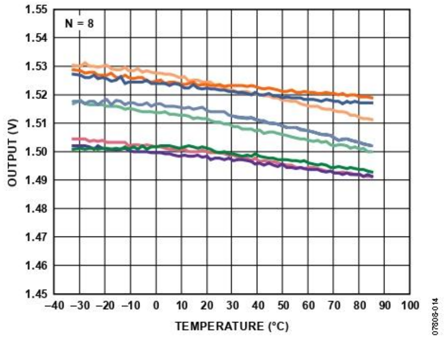


Figure 12. X-Axis Zero g Bias vs. Temperature—Eight Parts Soldered to PCB

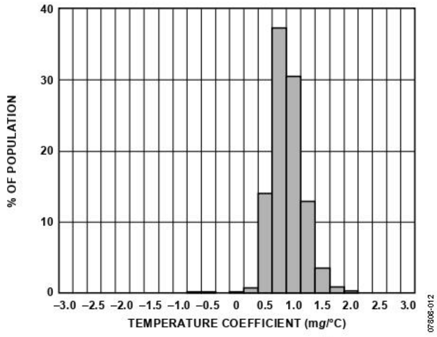


Figure 10. Y-Axis Zero g Bias Temperature Coefficient, $V_S = 3$ V

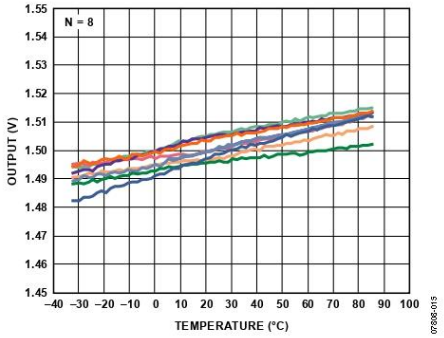


Figure 13. Y-Axis Zero g Bias vs. Temperature—Eight Parts Soldered to PCB

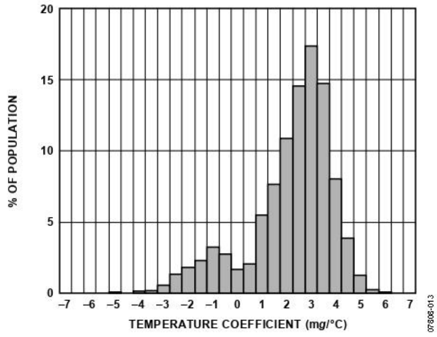


Figure 11. Z-Axis Zero g Bias Temperature Coefficient, $V_S = 3$ V

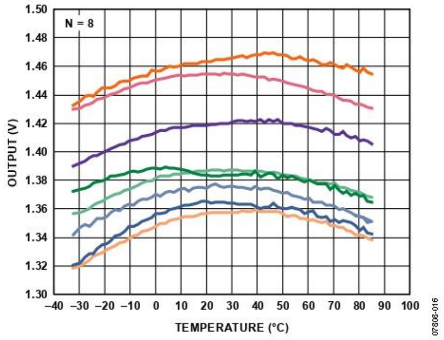


Figure 14. Z-Axis Zero g Bias vs. Temperature—Eight Parts Soldered to PCB

ADXL335

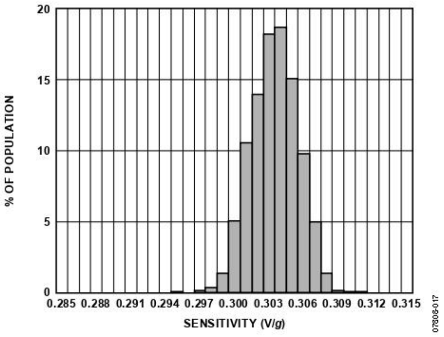


Figure 15. X-Axis Sensitivity at 25°C, V_S = 3 V

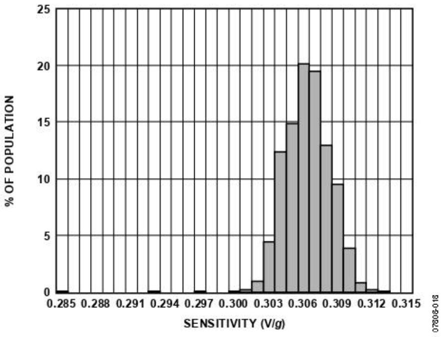


Figure 16. Y-Axis Sensitivity at 25°C, V_S = 3 V

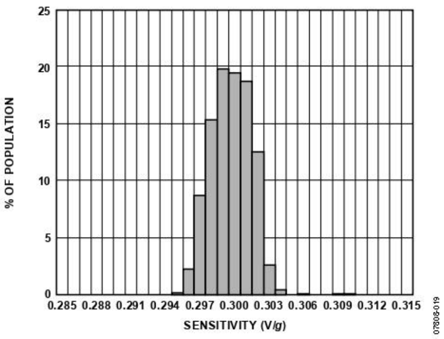


Figure 17. Z-Axis Sensitivity at 25°C, V_S = 3 V

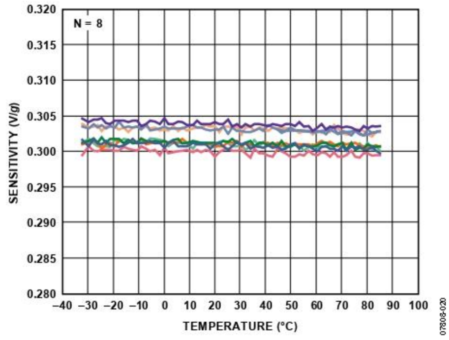


Figure 18. X-Axis Sensitivity vs. Temperature— Eight Parts Soldered to PCB, V_S = 3 V

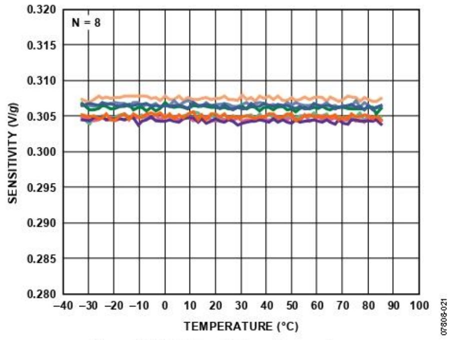


Figure 19. Y-Axis Sensitivity vs. Temperature— Eight Parts Soldered to PCB, V_S = 3 V

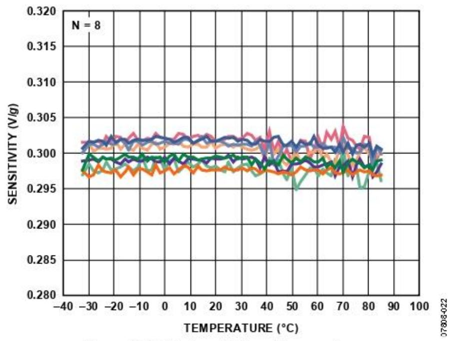


Figure 20. Z-Axis Sensitivity vs. Temperature— Eight Parts Soldered to PCB, V_S = 3 V

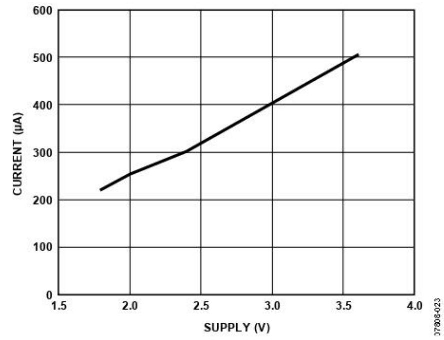


Figure 21. Typical Current Consumption vs. Supply Voltage

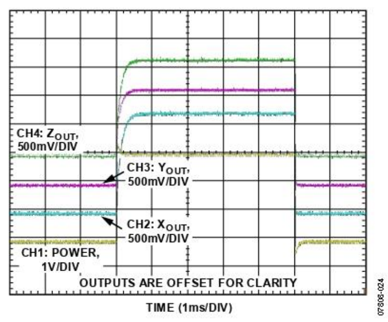


Figure 22. Typical Turn-On Time, $V_S = 3$ V

THEORY OF OPERATION

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ±3 *g* minimum. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The demodulator output is amplified and brought off-chip through a 32 kΩ resistor. The user then sets the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

MECHANICAL SENSOR

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes' sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity. Mechanical misalignment can, of course, be calibrated out at the system level.

PERFORMANCE

Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or nonmonotonic behavior, and temperature hysteresis is very low (typically less than 3 m*g* over the −25°C to +70°C temperature range).

APPLICATIONS INFORMATION

POWER SUPPLY DECOUPLING

For most applications, a single 0.1 µF capacitor, C_{DC}, placed close to the ADXL335 supply pins adequately decouples the accelerometer from noise on the power supply. However, in applications where noise is present at the 50 kHz internal clock frequency (or any harmonic thereof), additional care in power supply bypassing is required because this noise can cause errors in acceleration measurement.

If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite bead can be inserted in the supply line. Additionally, a larger bulk bypass capacitor (1 µF or greater) can be added in parallel to C_{DC}. Ensure that the connection from the ADXL335 ground to the power supply ground is low impedance because noise transmitted through ground has a similar effect to noise transmitted through V_S.

SETTING THE BANDWIDTH USING C_X, C_Y, AND C_Z

The ADXL335 has provisions for band limiting the X_{OUT}, Y_{OUT}, and Z_{OUT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is

$$F_{-3\,dB} = 1/(2\pi(32\ k\Omega) \times C_{(X, Y, Z)})$$

or more simply

$$F_{-3\,dB} = 5\ \mu F/C_{(X, Y, Z)}$$

The tolerance of the internal resistor (R_{FILT}) typically varies as much as ±15% of its nominal value (32 kΩ), and the bandwidth varies accordingly. A minimum capacitance of 0.0047 µF for C_X, C_Y, and C_Z is recommended in all cases.

Table 4. Filter Capacitor Selection, C_X, C_Y, and C_Z

Bandwidth (Hz)	Capacitor (µF)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

SELF-TEST

The ST pin controls the self-test feature. When this pin is set to V_S, an electrostatic force is exerted on the accelerometer beam. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output is −1.08 *g* (corresponding to −325 mV) in the X-axis, +1.08 *g* (or +325 mV) on the Y-axis, and +1.83 *g* (or +550 mV) on the Z-axis. This ST pin can be left open-circuit or connected to common (COM) in normal use.

Never expose the ST pin to voltages greater than V_S + 0.3 V. If this cannot be guaranteed due to the system design (for instance, if there are multiple supply voltages), then a low V_F clamping diode between ST and V_S is recommended.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The selected accelerometer bandwidth ultimately determines the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor to improve the resolution of the accelerometer. Resolution is dependent on the analog filter bandwidth at X_{OUT}, Y_{OUT}, and Z_{OUT}.

The output of the ADXL335 has a typical bandwidth of greater than 500 Hz. The user must filter the signal at this point to limit aliasing errors. The analog bandwidth must be no more than half the analog-to-digital sampling frequency to minimize aliasing. The analog bandwidth can be further decreased to reduce noise and improve resolution.

The ADXL335 noise has the characteristics of white Gaussian noise, which contributes equally at all frequencies and is described in terms of µ*g*/√Hz (the noise is proportional to the square root of the accelerometer bandwidth). The user should limit bandwidth to the lowest frequency needed by the application to maximize the resolution and dynamic range of the accelerometer.

With the single-pole, roll-off characteristic, the typical noise of the ADXL335 is determined by

$$rms\ Noise = Noise\ Density \times (\sqrt{BW \times 1.6})$$

It is often useful to know the peak value of the noise. Peak-to-peak noise can only be estimated by statistical methods. Table 5 is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table 5. Estimation of Peak-to-Peak Noise

Peak-to-Peak Value	% of Time That Noise Exceeds Nominal Peak-to-Peak Value
2 × rms	32
4 × rms	4.6
6 × rms	0.27
8 × rms	0.006

USE WITH OPERATING VOLTAGES OTHER THAN 3 V

The ADXL335 is tested and specified at V_S = 3 V; however, it can be powered with V_S as low as 1.8 V or as high as 3.6 V. Note that some performance parameters change as the supply voltage is varied.

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At V_S = 3.6 V, the output sensitivity is typically 360 mV/*g*. At V_S = 2 V, the output sensitivity is typically 195 mV/*g*.

The zero *g* bias output is also ratiometric, thus the zero *g* output is nominally equal to $V_S/2$ at all supply voltages.

The output noise is not ratiometric but is absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/*g*) increases while the noise voltage remains constant. At V_S = 3.6 V, the X-axis and Y-axis noise density is typically 120 µ*g*/√Hz, whereas at V_S = 2 V, the X-axis and Y-axis noise density is typically 270 µ*g*/√Hz.

Self-test response in *g* is roughly proportional to the square of the supply voltage. However, when ratiometricity of sensitivity is factored in with supply voltage, the self-test response in volts is roughly proportional to the cube of the supply voltage. For example, at V_S = 3.6 V, the self-test response for the ADXL335 is approximately −560 mV for the X-axis, +560 mV for the Y-axis, and +950 mV for the Z-axis.

At V_S = 2 V, the self-test response is approximately −96 mV for the X-axis, +96 mV for the Y-axis, and −163 mV for the Z-axis.

The supply current decreases as the supply voltage decreases. Typical current consumption at V_S = 3.6 V is 375 µA, and typical current consumption at V_S = 2 V is 200 µA.

AXES OF ACCELERATION SENSITIVITY

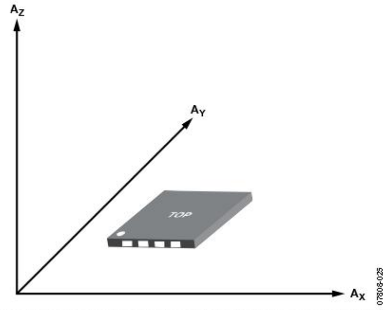


Figure 23. Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis.

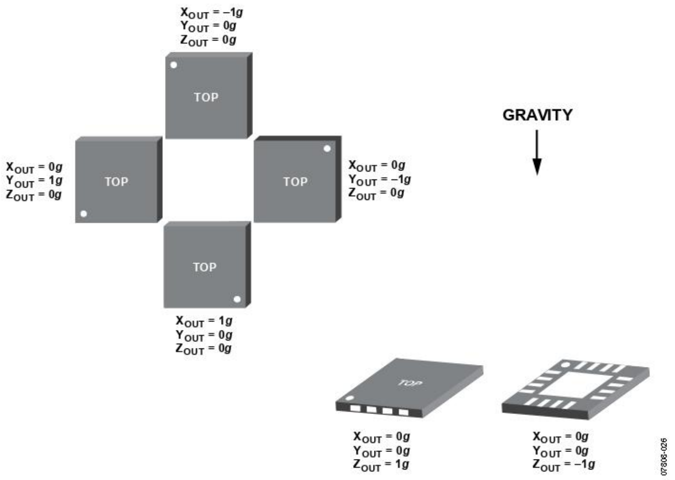


Figure 24. Output Response vs. Orientation to Gravity

LAYOUT AND DESIGN RECOMMENDATIONS

The recommended soldering profile is shown in Figure 25 followed by a description of the profile features in Table 6. The recommended PCB layout or solder land drawing is shown in Figure 26.

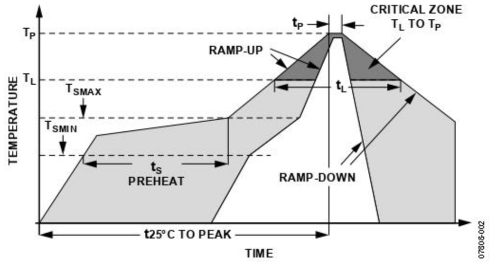


Figure 25. Recommended Soldering Profile

Table 6. Recommended Soldering Profile

Profile Feature	Sn63/Pb37	Pb-Free
Average Ramp Rate (T_L to T_P)	3°C/sec max	3°C/sec max
Preheat		
Minimum Temperature (T_{SMIN})	100°C	150°C
Maximum Temperature (T_{SMAX})	150°C	200°C
Time (T_{SMIN} to T_{SMAX})(t_S)	60 sec to 120 sec	60 sec to 180 sec
T_{SMAX} to T_L		
Ramp-Up Rate	3°C/sec max	3°C/sec max
Time Maintained Above Liquidous (T_L)		
Liquidous Temperature (T_L)	183°C	217°C
Time (t_L)	60 sec to 150 sec	60 sec to 150 sec
Peak Temperature (T_P)	240°C + 0°C/−5°C	260°C + 0°C/−5°C
Time Within 5°C of Actual Peak Temperature (t_P)	10 sec to 30 sec	20 sec to 40 sec
Ramp-Down Rate	6°C/sec max	6°C/sec max
Time 25°C to Peak Temperature	6 minutes max	8 minutes max

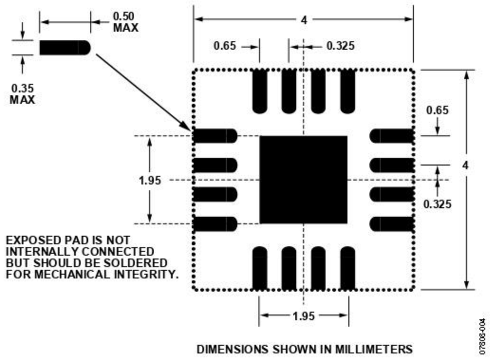


Figure 26. Recommended PCB Layout

ADXL335

OUTLINE DIMENSIONS

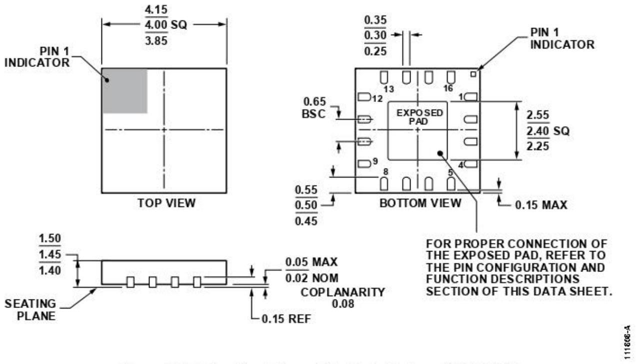


Figure 27. 16-Lead Lead Frame Chip Scale Package [LFCSP_LQ]
4 mm × 4 mm Body, 1.45 mm Thick Quad
(CP-16-14)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Measurement Range	Specified Voltage	Temperature Range	Package Description	Package Option
ADXL335BCPZ[1]	±3 *g*	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ–RL[1]	±3 *g*	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ–RL7[1]	±3 *g*	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
EVAL-ADXL335Z[1]				Evaluation Board	

[1] Z = RoHS Compliant Part.

NOTES

NOTES

Annex III: *Datasheet* de l'AT B5

LEM

AC Current Transducer AT-B5

Split-core transducer for the electronic measurement of AC waveform currents, with galvanic separation between the primary circuit and the secondary circuit. 0-5 V DC switch voltage output proportional to the RMS value of the primary current.

I_{PN} = 5 ... 150 A

Electrical data

Primary nominal RMS current I_{PN} (At RMS)	Output voltage V_{out} (V DC)	Type
5	0-5	AT 5 B5
10	0-5	AT 10 B5
20	0-5	AT 20 B5
50	0-5	AT 50 B5
100	0-5	AT 100 B5
150	0-5	AT 150 B5

R_L	Load resistance		≥ 1	MΩ
U_C	Supply voltage		self-powered	
V_{SZ}	Output clamping voltage		7.5	V
$\hat{I}_P$	Overload capability	- continuous	120	% of I_{PN}
		- 1 min	150	% of I_{PN}

Performance data

X	Accuracy @ I_{PN}, T_A = 25 °C (excluding offset)	< ±1.5	% of I_{PN}
ε_L	Linearity error	< ±0.5	% of I_{PN}
t_r	Step response time to 90 % of I_{PN}	≤ 300	ms
BW	Frequency bandwidth	50/60	Hz

General data

T_A	Ambient operating temperature	−20 … +60	°C
T_S	Ambient storage temperature	−20 … +85	°C
m	Mass	90	g
IPxx	Protection degree	IP 40	

Note: Deviation of the output during test IEC 61000-4-3 @ 10 V/m between 280 to 500 MHz.

Features

- RMS (average) output
- 0-5 V DC voltage output
- Split-core type
- ⌀ 16 mm sensing aperture for non-contact measurement
- Terminal output
- Insulating plastic case recognized according to UL 94-V0.

Advantages

- High insulation between primary and secondary circuits
- Compact case
- Cost-effective solution
- Easy installation.

Applications

- **Automation and Supervision**
  Current measurement for process regulation by distributed PLCs or remote control (e.g. SCADA software)
- **Safety and Condition Monitoring**
  Load monitoring for protection systems and predictive maintenance (e.g. conveyers, pumps or HVAC motors).

Application domain

- Energy and Automation.

N° U2.05.08.100.0; U2.05.13.100.0; U2.05.17.100.0; U2.05.25.100.0; U2.05.34.100.0; U2.05.39.100.0

Page 1/4

9May2018/Version 8 LEM reserves the right to carry out modifications on its transducers, in order to improve them, without prior notice www.lem.com

LEM

Current Transducer AT-B5

Insulation coordination

U_b	Rated insulation RMS voltage [1], reinforced or basic insulation with IEC 61010-1 standards and following conditions: - Reinforced insulation - Over voltage category CAT III - Pollution degree PD2 - Heterogeneous field	300	V
U_d	RMS voltage for AC insulation test [2], 50 Hz, 1 min	3.5 Min	kV
d_{Cp}	Creepage distance	6	mm
d_{Cl}	Clearance	6	mm
CTI	Comparative tracking index (group I)	600	

Notes: [1] If insulated cable is used for the primary circuit, the voltage category could be improved according to the insulation coordination given by the cable manufacturer

[2] Between primary (completely filling the hole) and secondary.

LEM

Safety and warning notes

In order to guarantee safe operation of the transducer and to be able to make proper use of all features and functions, please read these instructions thoroughly! Safe operation can only be guaranteed if the transducer is used for the purpose it has been designed for and within the limits of the technical specifications. Ensure you get up-to-date technical information that can be found in the latest associated datasheet under www.lem.com.

Caution! Risk of danger

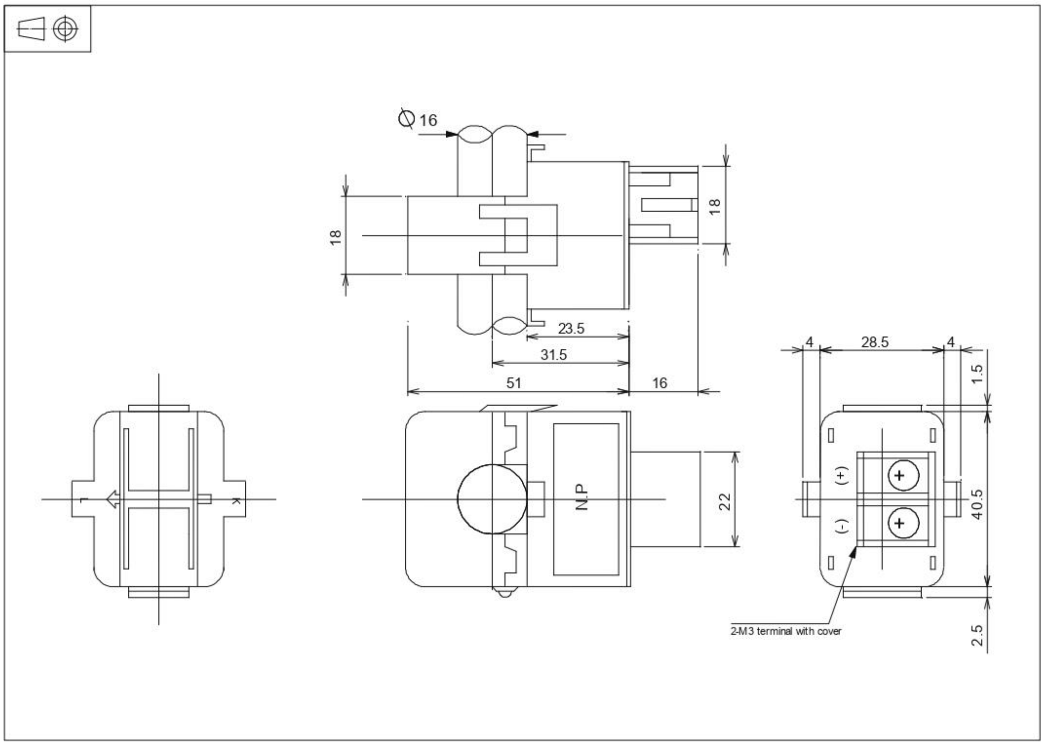
Ignoring the warnings can lead to serious injury and/or cause damage!
The electric measuring transducer may only be installed and put into operation by qualified personnel that have received an appropriate training.
The corresponding national regulations shall be observed during installation and operation of the transducer and any electrical conductor.
The transducer shall be used in electric/electronic equipment with respect to applicable standards and safety requirements and in accordance with all the related systems and components manufacturer' operating instructions.

Caution, Risk of electrical shock

When operating the transducer, certain parts of the module may carry hazardous live voltage (eg. primary conductor, power supply).
The user shall ensure to take all measures necessary to protect against electrical shock. The transducer is a build-in device containing conducting parts that shall not be accessible after installation.
A protective enclosure or additional insulation barrier may be necessary.
The transducer shall not be put into operation if the jaw opening is open (split core version) or the installation is not completed.
Installation and maintenance shall be done with the main power supply disconnected except if there are no hazardous live parts in or in close proximity to the system and if the applicable national regulations are fully observed.

Safe and trouble-free operation of this transducer can only be guaranteed if transport, storage and installation are carried out correctly and operation and maintenance are carried out with care.

Dimensions AT-B5 (in mm)

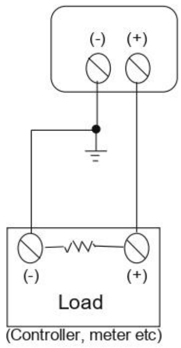


Mechanical characteristics

- General tolerance ±1 mm
- Primary aperture Ø 16 mm
- Fastening Cable tie

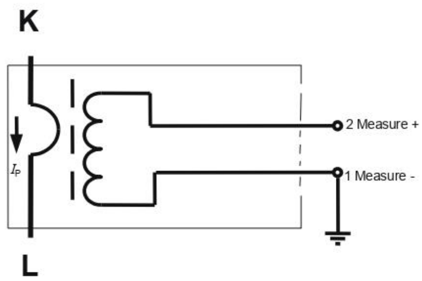
Connections

- Wires up to 2 mm Ø



Remarks

- Attention: contact areas (air gap) must be kept clean (particle free) to ensure proper performance.



Annex IV: *Datasheet* del CNY-70

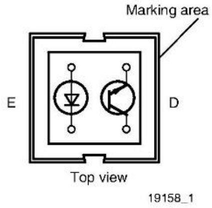
VISHAY. www.vishay.com

CNY70

Vishay Semiconductors

Reflective Optical Sensor with Transistor Output

21835



DESCRIPTION

The CNY70 is a reflective sensor that includes an infrared emitter and phototransistor in a leaded package which blocks visible light.

FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 7 x 7 x 6
- Peak operating distance: < 0.5 mm
- Operating range within > 20 % relative collector current: 0 mm to 5 mm
- Typical output current under test: I_C = 1 mA
- Emitter wavelength: 950 nm
- Daylight blocking filter
- Lead (Pb)-free soldering released
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

Pb-free e4 RoHS COMPLIANT

APPLICATIONS

- Optoelectronic scanning and switching devices i.e., index sensing, coded disk scanning etc. (optoelectronic encoder assemblies).

PRODUCT SUMMARY				
PART NUMBER	DISTANCE FOR MAXIMUM CTR_{rel} (1) (mm)	DISTANCE RANGE FOR RELATIVE I_{out} > 20 % (mm)	TYPICAL OUTPUT CURRENT UNDER TEST (2) (mA)	DAYLIGHT BLOCKING FILTER INTEGRATED
CNY70	0	0 to 5	1	Yes

Notes

(1) CTR: current transfere ratio, I_{out}/I_{in}

(2) Conditions like in table basic charactristics/sensors

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	VOLUME (1)	REMARKS
CNY70	Tube	MOQ: 4000 pcs, 80 pcs/tube	-

Note

(1) MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS (T_{amb} = 25 °C, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
COUPLER				
Total power dissipation	$T_{amb} \leq 25$ °C	P_{tot}	200	mW
Ambient temperature range		T_{amb}	- 40 to + 85	°C
Storage temperature range		T_{stg}	- 40 to + 100	°C
Soldering temperature	Distance to case 2 mm, t £ 5 s	T_{sd}	260	°C
INPUT (EMITTER)				
Reverse voltage		V_R	5	V
Forward current		I_F	50	mA
Forward surge current	$t_p \leq 10$ µs	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25$ °C	P_V	100	mW
Junction temperature		T_j	100	°C

Rev. 1.8, 30-Jul-12 1 Document Number: 83751

For technical questions, contact: sensorstechsupport@vishay.com

ABSOLUTE MAXIMUM RATINGS (T_{amb} = 25 °C, unless otherwise specified)				
PARAMETER	**TEST CONDITION**	**SYMBOL**	**VALUE**	**UNIT**
OUTPUT (DETECTOR)				
Collector emitter voltage		V_{CEO}	32	V
Emitter collector voltage		V_{ECO}	7	V
Collector current		I_C	50	mA
Power dissipation	$T_{amb} \leq 25$ °C	P_V	100	mW
Junction temperature		T_j	100	°C

ABSOLUTE MAXIMUM RATINGS

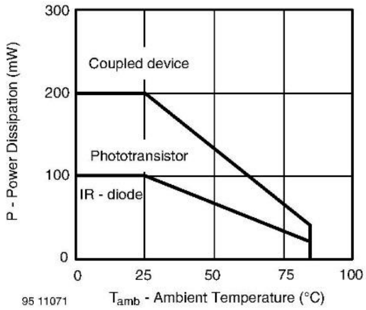


Fig. 1 - Power Dissipation vs. Ambient Temperature

BASIC CHARACTERISTICS (T_{amb} = 25 °C, unless otherwise specified)						
PARAMETER	**TEST CONDITION**	**SYMBOL**	**MIN.**	**TYP.**	**MAX.**	**UNIT**
COUPLER						
Collector current	V_{CE} = 5 V, I_F = 20 mA, d = 0.3 mm (figure 1)	I_C [2]	0.3	1.0		mA
Cross talk current	V_{CE} = 5 V, I_F = 20 mA, (figure 2)	I_{CX} [3]			600	nA
Collector emitter saturation voltage	I_F = 20 mA, I_C = 0.1 mA, d = 0.3 mm (figure 1)	V_{CEsat} [2]			0.3	V
INPUT (EMITTER)						
Forward voltage	I_F = 50 mA	V_F		1.25	1.6	V
Radiant intensity	I_F = 50 mA, t_p = 20 ms	I_e			7.5	mW/sr
Peak wavelength	I_F = 100 mA	λ_P	940			nm
Virtual source diameter	Method: 63 % encircled energy	d		1.2		mm
OUTPUT (DETECTOR)						
Collector emitter voltage	I_C = 1 mA	V_{CEO}	32			V
Emitter collector voltage	I_E = 100 μA	V_{ECO}	5			V
Collector dark current	V_{CE} = 20 V, I_F = 0 A, E = 0 lx	I_{CEO}			200	nA

Notes

[1] Measured with the "Kodak neutral test card", white side with 90 % diffuse reflectance

[2] Measured without reflecting medium

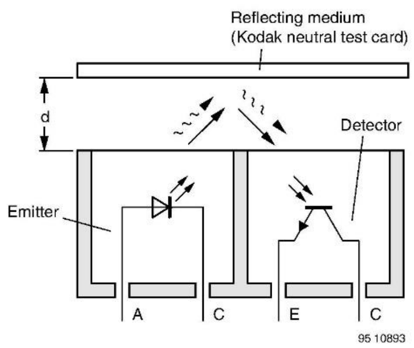


Fig. 2 - Test Condition

BASIC CHARACTERISTICS (T_{amb} = 25 °C, unless otherwise specified)

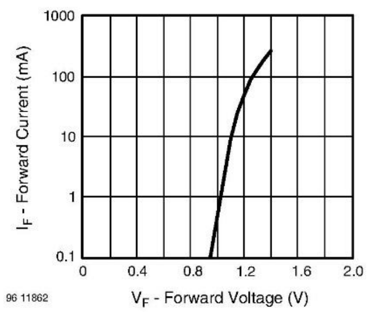


Fig. 3 - Forward Current vs. Forward Voltage

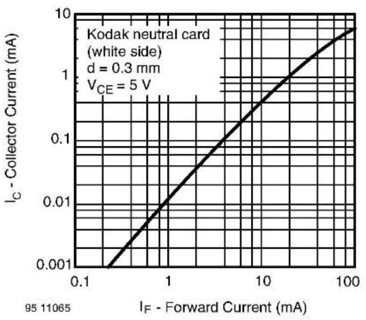


Fig. 5 - Collector Current vs. Forward Current

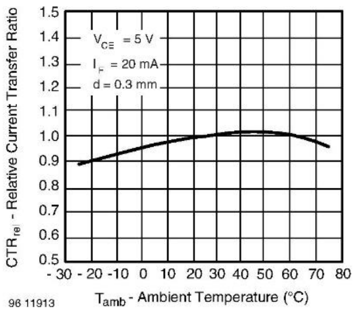


Fig. 4 - Relative Current Transfer Ratio vs. Ambient Temperature

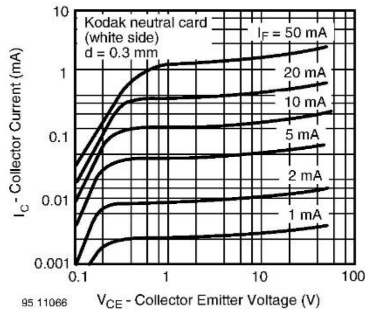


Fig. 6 - Collector Current vs. Collector Emitter Voltage

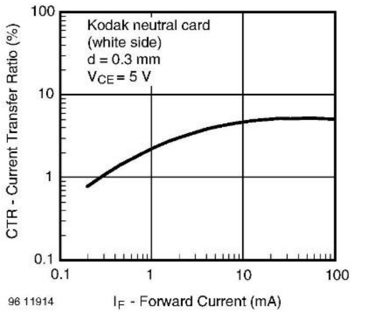


Fig. 7 - Current Transfer Ratio vs. Forward Current

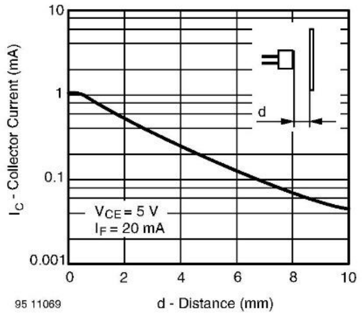


Fig. 9 - Collector Current vs. Distance

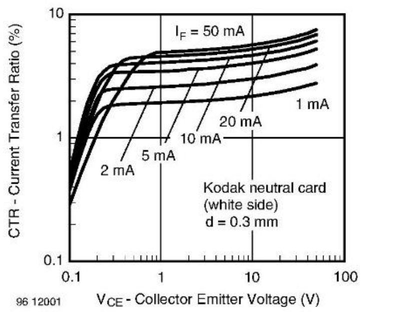


Fig. 8 - Current Transfer Ratio vs. Collector Emitter Voltage

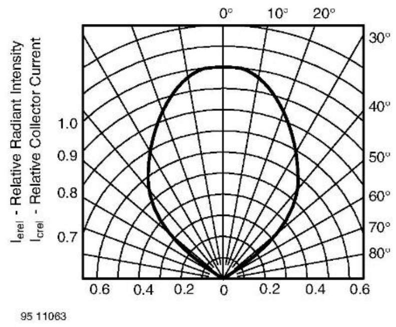


Fig. 10 - Relative Radiant Intensity/Collector Current vs. Angular Displacement

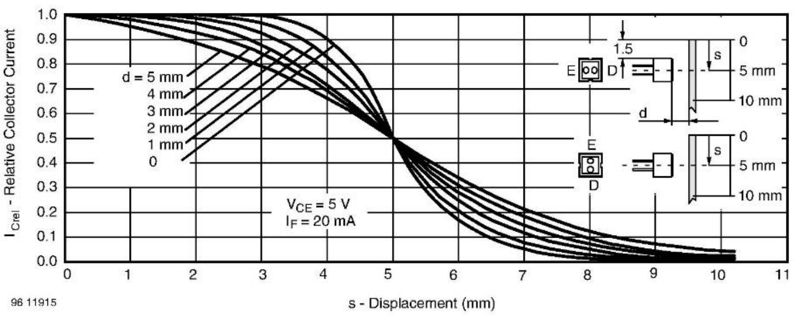
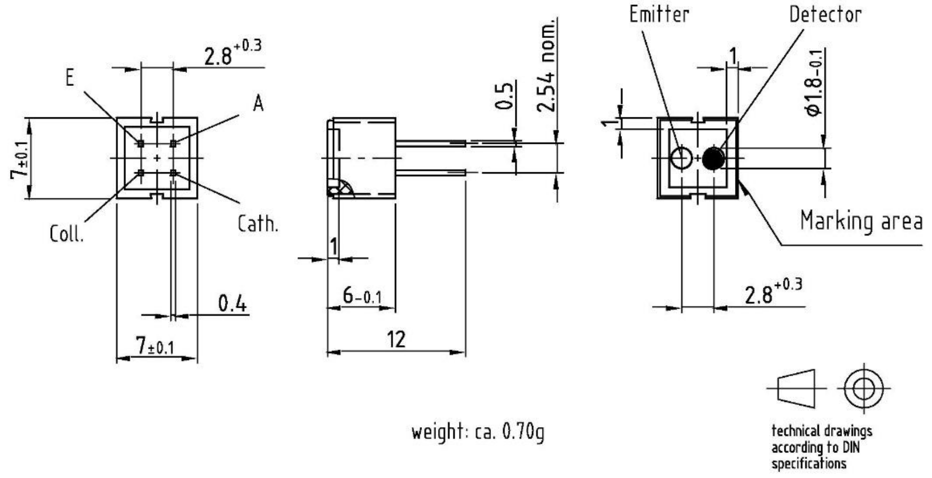


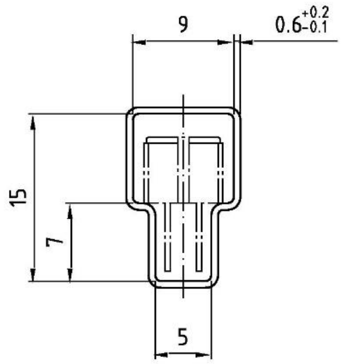
Fig. 11 - Relative Collector Current vs. Displacement

PACKAGE DIMENSIONS in millimeters



Drawing-No.: 6.544-5062.01-4
Issue: 6; 03.05.06
95 11345

TUBE DIMENSIONS in millimeters



With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm

Drawing-No.: 9.700-5097.01-4
Issue: 1; 25.02.00
20291

Packaging and Ordering Information

PART NUMBER	MOQ (1)	PCS PER TUBE	TUBE SPEC. (FIGURE)	CONSTITUENTS (FORMS)
CNY70	4000	80	1	28
TCPT1300X01	2000	Reel	(2)	29
TCRT1000	1000	Bulk	-	26
TCRT1010	1000	Bulk	-	26
TCRT5000	4500	50	2	27
TCRT5000L	2400	48	3	27
TCST1030	5200	65	5	24
TCST1030L	2600	65	6	24
TCST1103	1020	85	4	24
TCST1202	1020	85	4	24
TCST1230	4800	60	7	24
TCST1300	1020	85	4	24
TCST2103	1020	85	4	24
TCST2202	1020	85	4	24
TCST2300	1020	85	4	24
TCST5250	4860	30	8	24
TCUT1300X01	2000	Reel	(2)	29
TCZT8020-PAER	2500	Bulk	-	22

Notes
(1) MOQ: minimum order quantity
(2) Please refer to datasheets

TUBE SPECIFICATION FIGURES

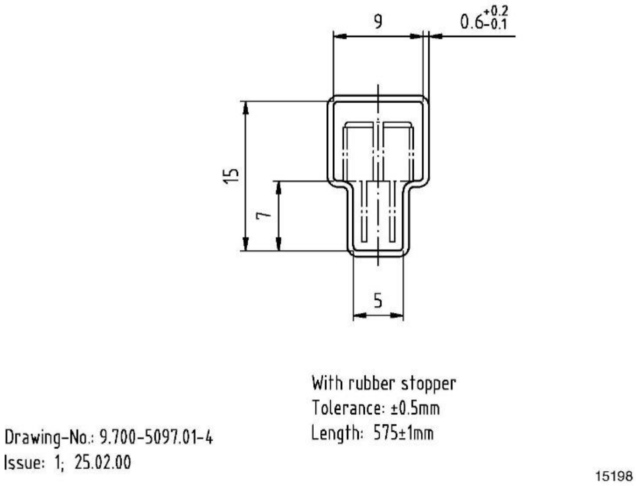


Fig. 1

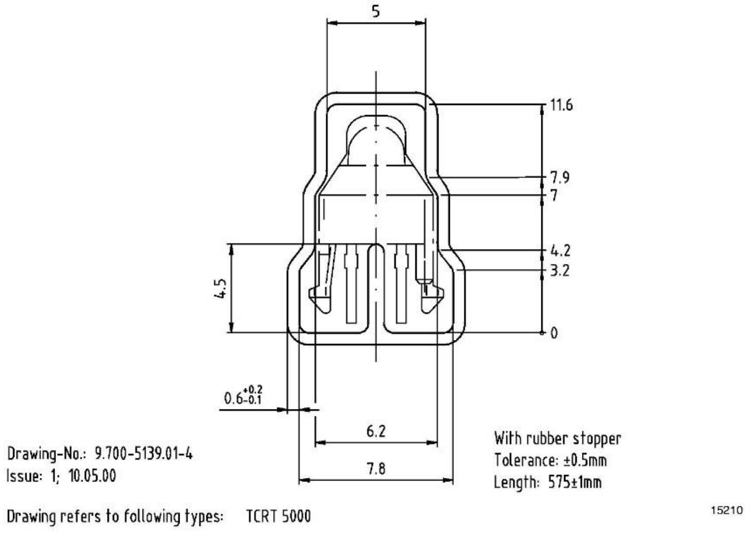

5
11.6
7.9
7
4.2
3.2
0
4.5
0.6+0.2/-0.1
6.2
7.8
Drawing-No.: 9.700-5139.01-4
Issue: 1; 10.05.00
With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
Drawing refers to following types: TCRT 5000
15210

Fig. 2

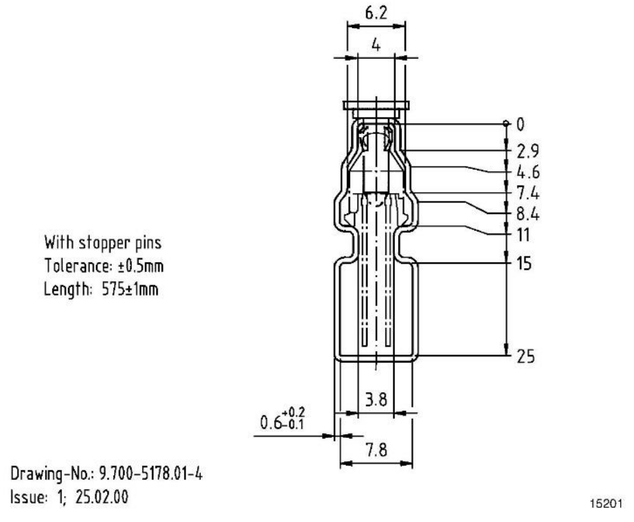

6.2
4
0
2.9
4.6
7.4
8.4
11
15
25
With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm
3.8
0.6+0.2/-0.1
7.8
Drawing-No.: 9.700-5178.01-4
Issue: 1; 25.02.00
15201

Fig. 3

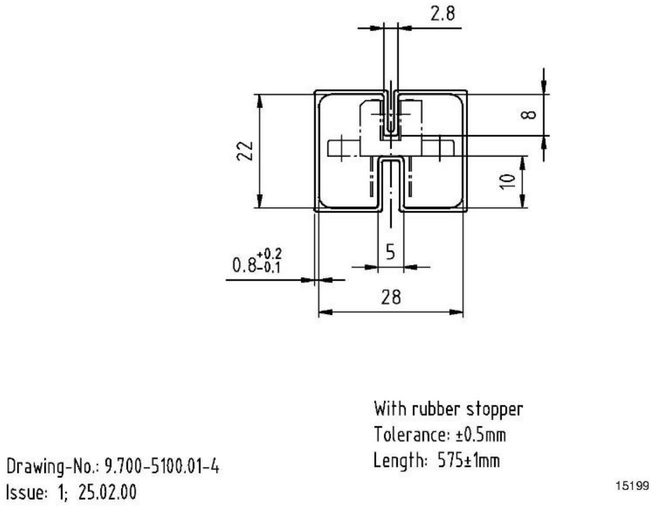

2.8
8
22
10
0.8+0.2 -0.1
5
28
With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
Drawing-No.: 9.700-5100.01-4
Issue: 1; 25.02.00
15199

Fig. 4

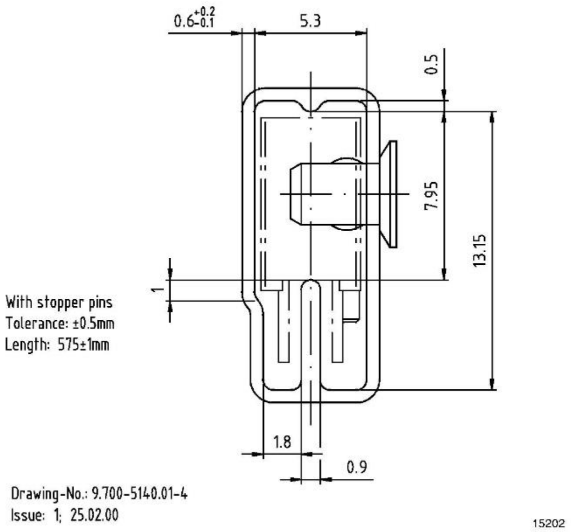

0.6+0.2 -0.1
5.3
0.5
7.95
13.15
1
With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm
1.8
0.9
Drawing-No.: 9.700-5140.01-4
Issue: 1; 25.02.00
15202

Fig. 5

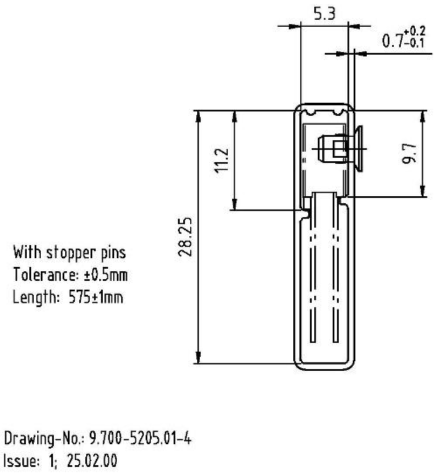

5.3
0.7+0.2 -0.1
11.2
9.7
28.25
With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm
Drawing-No.: 9.700-5205.01-4
Issue: 1; 25.02.00
15196

Fig. 6

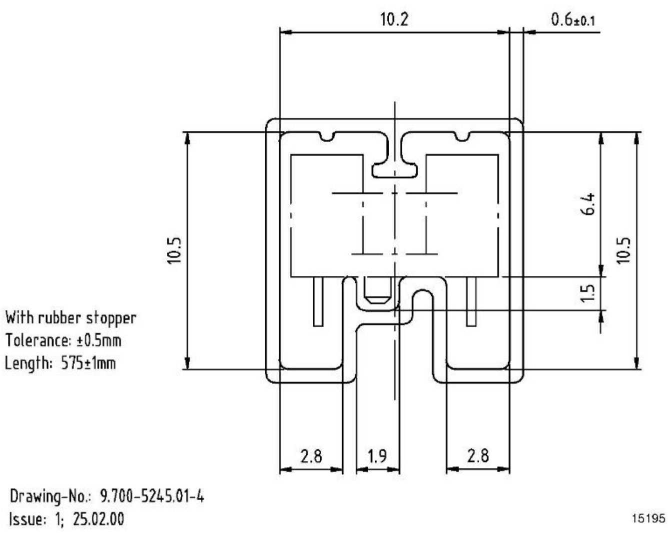

10.2
0.6±0.1
10.5
6.4
10.5
1.5
With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
2.8
1.9
2.8
Drawing-No.: 9.700-5245.01-4
Issue: 1; 25.02.00
15195

Fig. 7

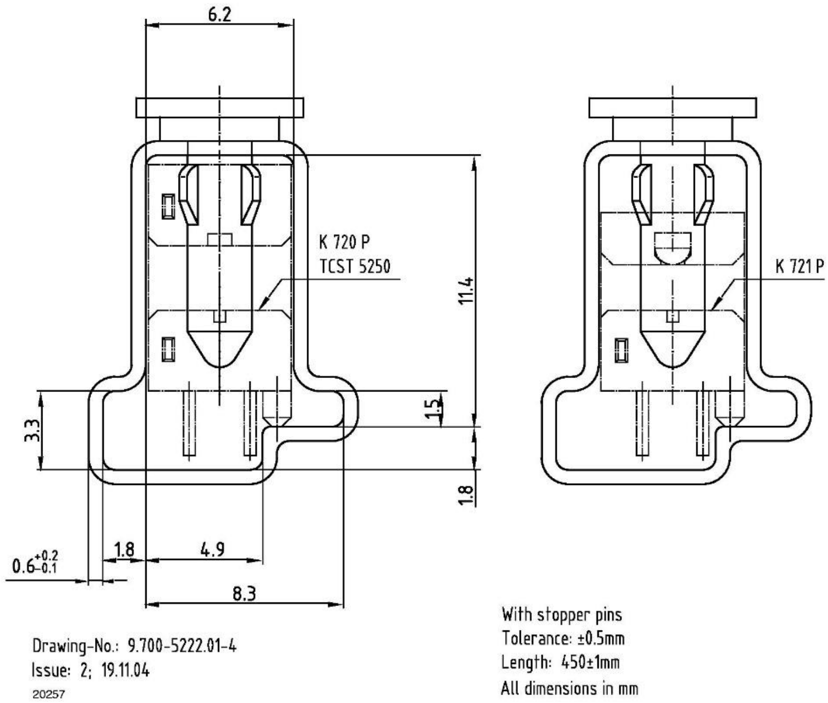

6.2
K 720 P
TCST 5250
11.4
K 721 P
3.3
1.5
1.8
0.6$^{+0.2}_{-0.1}$
1.8
4.9
8.3
Drawing-No.: 9.700-5222.01-4
Issue: 2; 19.11.04
20257
With stopper pins
Tolerance: ±0.5mm
Length: 450±1mm
All dimensions in mm

Fig. 8

VISHAY. www.vishay.com

Legal Disclaimer Notice

Vishay

Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, “Vishay”), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay’s knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer’s responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and / or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer’s technical experts. Product specifications do not expand or otherwise modify Vishay’s terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Annex V: *Datasheet* del DS18B20

*Click **here** for production status of specific part numbers.*

DS18B20

Programmable Resolution 1-Wire Digital Thermometer

General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line (“parasite power”), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

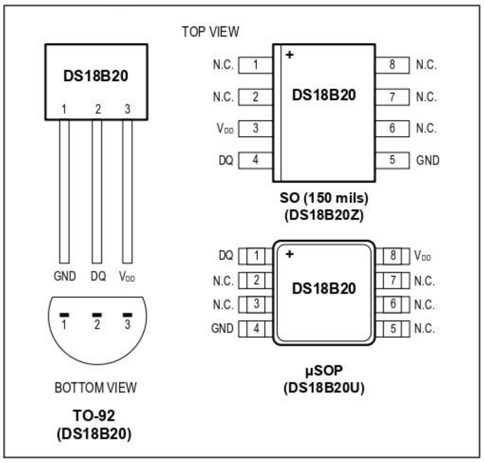
***Ordering Information* appears at end of data sheet.**

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
  - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
  - ±0.5°C Accuracy from -10°C to +85°C
  - Programmable Resolution from 9 Bits to 12 Bits
  - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
  - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages

Pin Configurations



19-7487; Rev 5; 9/18

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground ....-0.5V to +6.0V
Operating Temperature Range .... -55°C to +125°C
Storage Temperature Range .... -55°C to +125°C
Solder Temperature ....Refer to the IPC/JEDEC J-STD-020 Specification.

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Supply Voltage	V_{DD}	Local power (Note 1)		+3.0		+5.5	V
Pullup Supply Voltage	V_{PU}	Parasite power	(Notes 1, 2)	+3.0		+5.5	V
		Local power		+3.0		V_{DD}	
Thermometer Error	t_{ERR}	-10°C to +85°C	(Note 3)			±0.5	°C
		-30°C to +100°C				±1	
		-55°C to +125°C				±2	
Input Logic-Low	V_{IL}	(Notes 1, 4, 5)		-0.3		+0.8	V
Input Logic-High	V_{IH}	Local power	(Notes 1,6)	+2.2		The lower of 5.5 or V_{DD} + 0.3	V
		Parasite power		+3.0			
Sink Current	I_L	$V_{I/O}$ = 0.4V		4.0			mA
Standby Current	I_{DDS}	(Notes 7, 8)			750	1000	nA
Active Current	I_{DD}	V_{DD} = 5V (Note 9)			1	1.5	mA
DQ Input Current	I_{DQ}	(Note 10)			5		µA
Drift		(Note 11)			±0.2		°C

Note 1: All voltages are referenced to ground.
Note 2: The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$.
Note 3: See typical performance curve in Figure 1. Thermometer Error limits are 3-sigma values.
Note 4: Logic-low voltages are specified at a sink current of 4mA.
Note 5: To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
Note 6: Logic-high voltages are specified at a source current of 1mA.
Note 7: Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
Note 8: To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} – 0.3V ≤ DQ ≤ V_{DD}.
Note 9: Active current refers to supply current during active temperature conversions or EEPROM writes.
Note 10: DQ line is high ("high-Z" state).
Note 11: Drift data is based on a 1000-hour stress test at +125°C with V_{DD} = 5.5V.

AC Electrical Characteristics–NV Memory

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{WR}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

AC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Temperature Conversion Time	t_{CONV}	9-bit resolution	(Note 12)			93.75	ms
		10-bit resolution				187.5	
		11-bit resolution				375	
		12-bit resolution				750	
Time to Strong Pullup On	t_{SPON}	Start convert T command issued				10	µs
Time Slot	t_{SLOT}	(Note 12)		60		120	µs
Recovery Time	t_{REC}	(Note 12)		1			µs
Write 0 Low Time	t_{LOW0}	(Note 12)		60		120	µs
Write 1 Low Time	t_{LOW1}	(Note 12)		1		15	µs
Read Data Valid	t_{RDV}	(Note 12)				15	µs
Reset Time High	t_{RSTH}	(Note 12)		480			µs
Reset Time Low	t_{RSTL}	(Notes 12, 13)		480			µs
Presence-Detect High	t_{PDHIGH}	(Note 12)		15		60	µs
Presence-Detect Low	t_{PDLOW}	(Note 12)		60		240	µs
Capacitance	$C_{IN/OUT}$					25	pF

Note 12: See the timing diagrams in Figure 2.
Note 13: Under parasite power, if t_{RSTL} > 960µs, a power-on reset can occur.

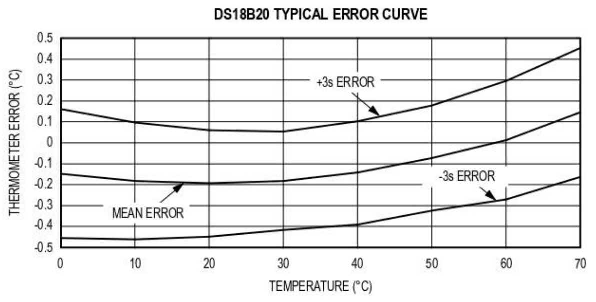


Figure 1. Typical Performance Curve

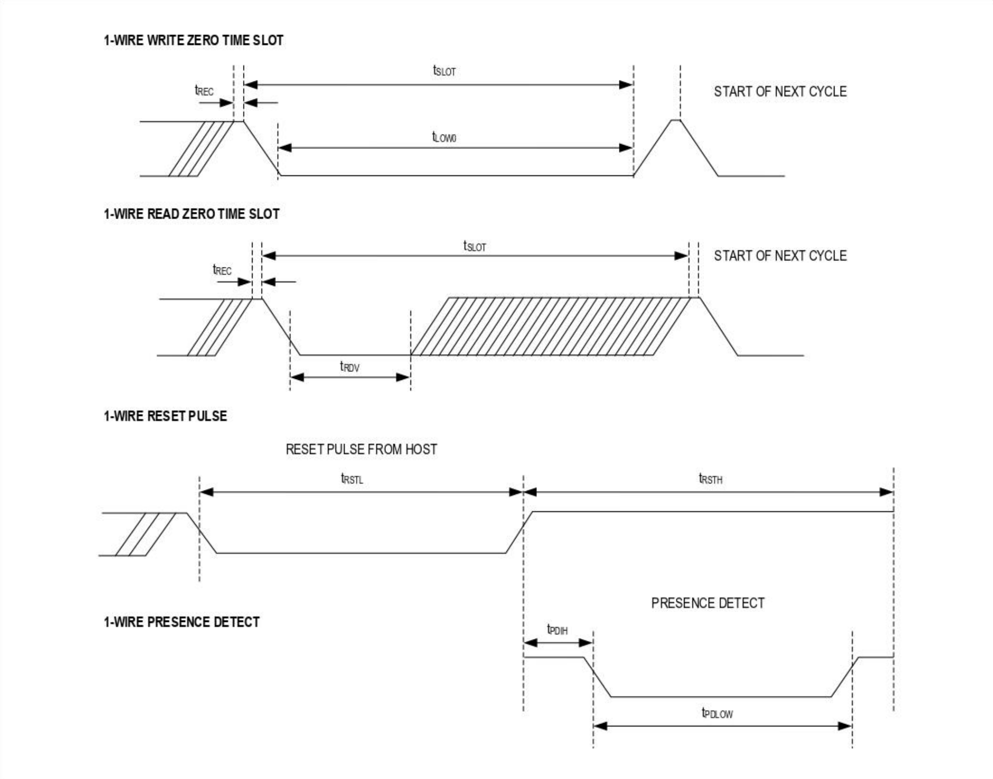


Figure 2. Timing Diagrams

Pin Description

PIN			NAME	FUNCTION
SO	µSOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	No Connection
3	8	3	V_{DD}	Optional V_{DD}. V_{DD} must be grounded for operation in parasite power mode.
4	1	2	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the *Powering the DS18B20* section.)
5	4	1	GND	Ground

Overview

Figure 3 shows a block diagram of the DS18B20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H, T_L, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-Wire Bus System* section.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor through the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18B20 may also be powered by an external supply on V_{DD}.

Operation—Measuring Temperature

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" (see the *1-Wire Bus System* section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *Powering the DS18B20* section.

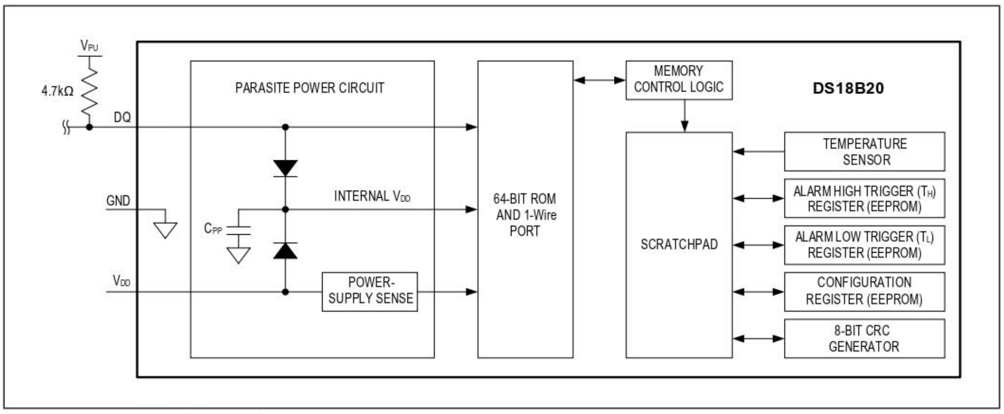


Figure 3. DS18B20 Block Diagram

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 4). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. Table 1 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

Operation—Alarm Signaling

After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 5). The sign bit (S) indicates if the value is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the *Memory* section.

Only bits 11 through 4 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the measured temperature is lower than

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Figure 4. Temperature Register Format

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

**The power-on reset value of the temperature register is +85°C.*

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Figure 5. T_H and T_L Register Format

or equal to T_L or higher than or equal to T_H, an alarm condition exists and an alarm flag is set inside the DS18B20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18B20s on the bus by issuing an Alarm Search [ECh] command. Any DS18B20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18B20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

Powering the DS18B20

The DS18B20 can be powered by an external supply on the V_{DD} pin, or it can operate in "parasite power" mode, which allows the DS18B20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 3 shows the DS18B20's parasite-power control circuitry, which "steals" power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18B20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low. When the DS18B20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

In parasite power mode, the 1-Wire bus and CPP can provide sufficient current to the DS18B20 for most operations as long as the specified timing and voltage requirements are met (see the *DC Electrical Characteristics* and *AC Electrical Characteristics*). However, when the DS18B20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by C_{PP}. To assure that the DS18B20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 6. The 1-Wire bus must be switched to the strong pullup within 10µs (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{CONV}) or data transfer (t_{WR} = 10ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18B20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 7. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

The use of parasite power is not recommended for temperatures above +100°C since the DS18B20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18B20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18B20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a "read time slot". During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

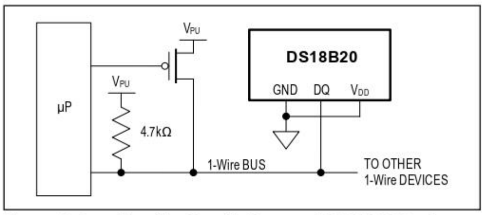


Figure 6. Supplying the Parasite-Powered DS18B20 During Temperature Conversions

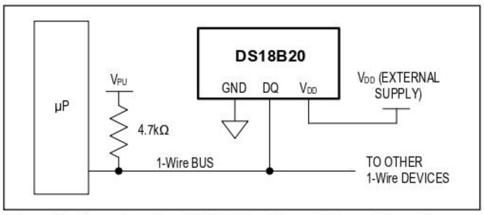


Figure 7. Powering the DS18B20 with an External Supply

64-BIT Lasered ROM code

Each DS18B20 contains a unique 64–bit code (see Figure 8) stored in ROM. The least significant 8 bits of the ROM code contain the DS18B20's 1-Wire family code: 28h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the *CRC Generation* section. The 64-bit ROM code and associated ROM function control logic allow the DS18B20 to operate as a 1-Wire device using the protocol detailed in the *1-Wire Bus System* section.

Memory

The DS18B20's memory is organized as shown in Figure 9. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L) and configuration register. Note that if the DS18B20 alarm function is not used, the TH and TL registers can serve as general-purpose memory. All memory commands are described in detail in the *DS18B20 Function Commands* section.

Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to TH and TL registers. Byte 4 contains the configuration register data, which is explained in detail in the *Configuration Register* section. Bytes 5, 6, and 7 are reserved for internal use by the device and cannot be overwritten.

Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 0 through 7 of the scratchpad. The DS18B20 generates this CRC using the method described in the *CRC Generation* section.

Data is written to bytes 2, 3, and 4 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18B20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H, T_L and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E^2 [B8h] command. The master can issue read time slots following the Recall E^2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

8-BIT CRC	48-BIT SERIAL NUMBER	8-BIT FAMILY CODE (28h)
MSB LSB	MSB LSB	MSB LSB

Figure 8. 64-Bit Lasered ROM Code

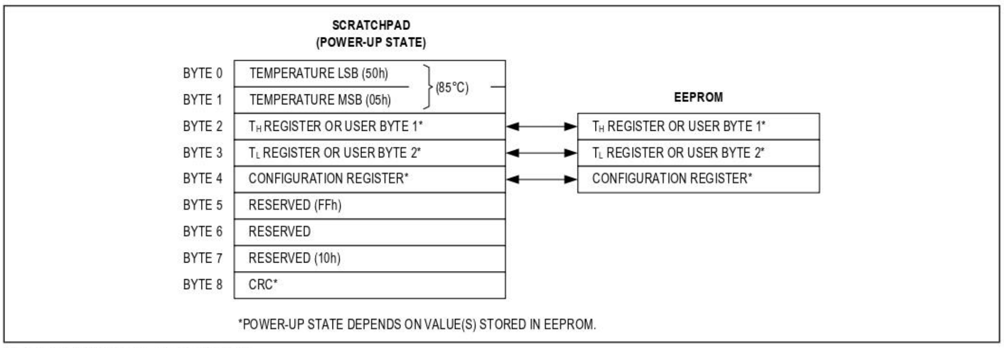


Figure 9. DS18B20 Memory Map

Configuration Register

Byte 4 of the scratchpad memory contains the configuration register, which is organized as illustrated in Figure 10. The user can set the conversion resolution of the DS18B20 using the R0 and R1 bits in this register as shown in Table 2. The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct tradeoff between resolution and conversion time. Bit 7 and bits 0 to 4 in the configuration register are reserved for internal use by the device and cannot be overwritten.

CRC Generation

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18B20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$CRC = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in Figure 11. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the recalculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18B20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Maxim 1-Wire cyclic redundancy check is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products*.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

Figure 10. Configuration Register

Table 2. Thermometer Resolution Configuration

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	($t_{CONV}/8$)
0	1	10	187.5ms	($t_{CONV}/4$)
1	0	11	375ms	($t_{CONV}/2$)
1	1	12	750ms	(t_{CONV})

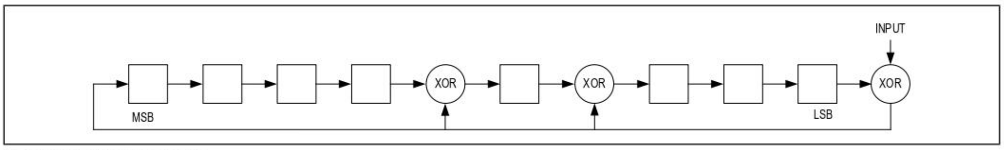


Figure 11. CRC Generator

1-Wire Bus System

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 is always a slave. When there is only one slave on the bus, the system is referred to as a "single-drop" system; the system is "multidrop" if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

Hardware Configuration

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open-drain or 3-state port. This allows each device to "release" the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 12.

The 1-Wire bus requires an external pullup resistor of approximately 5kΩ; thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480µs, all components on the bus will be reset.

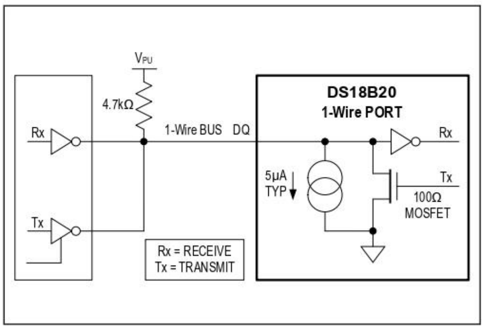


Figure 12. Hardware Configuration

Transaction Sequence

The transaction sequence for accessing the DS18B20 is as follows:

Step 1. Initialization

Step 2. ROM Command (followed by any required data exchange)

Step 3. DS18B20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18B20 is accessed, as the DS18B20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

Initialization

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18B20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the *1-Wire Signaling* section.

ROM Commands

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18B20 function command. A flowchart for operation of the ROM commands is shown in Figure 13.

Search Rom [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices.

If there is only one slave on the bus, the simpler Read ROM [33h] command can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to *Application Note 937: Book of iButton® Standards*. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

Read Rom [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

Match Rom [55H]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

Skip Rom [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18B20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

Alarm Search [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18B20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. See the *Operation—Alarm Signaling* section for an explanation of alarm flag operation.

DS18B20 Function Commands

After the bus master has used a ROM command to address the DS18B20 with which it wishes to communicate, the master can issue one of the DS18B20 function commands. These commands allow the master to write to and read from the DS18B20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18B20 function commands, which are described below, are summarized in Table 3 and illustrated by the flowchart in Figure 14.

Convert T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{CONV}) as described in the *Powering the DS18B20* section. If the DS18B20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18B20 will respond by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

Write Scratchpad [4Eh]

This command allows the master to write 3 bytes of data to the DS18B20's scratchpad. The first data byte is written into the T_H register (byte 2 of the scratchpad), the second byte is written into the T_L register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes MUST be written before the master issues a reset, or the data may be corrupted.

Read Scratchpad [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

iButton is a registered trademark of Maxim Integrated Products, Inc.

Copy Scratchpad [48h]

This command copies the contents of the scratchpad T_H, T_L and configuration registers (bytes 2, 3 and 4) to EEPROM. If the device is being used in parasite power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the *Powering the DS18B20* section.

Recall E2 [B8h]

This command recalls the alarm trigger values (T_H and T_L) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E^2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

Read Power Supply [B4h]

The master device issues this command followed by a read time slot to determine if any DS18B20s on the bus are using parasite power. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. See the *Powering the DS18B20* section for usage information for this command.

Table 3. DS18B20 Function Command Set

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H, T_L, and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies T_H, T_L, and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E^2	Recalls T_H, T_L, and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

Note 1: For parasite-powered DS18B20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
Note 2: The master can interrupt the transmission of data at any time by issuing a reset.
Note 3: All three bytes must be written before a reset is issued.

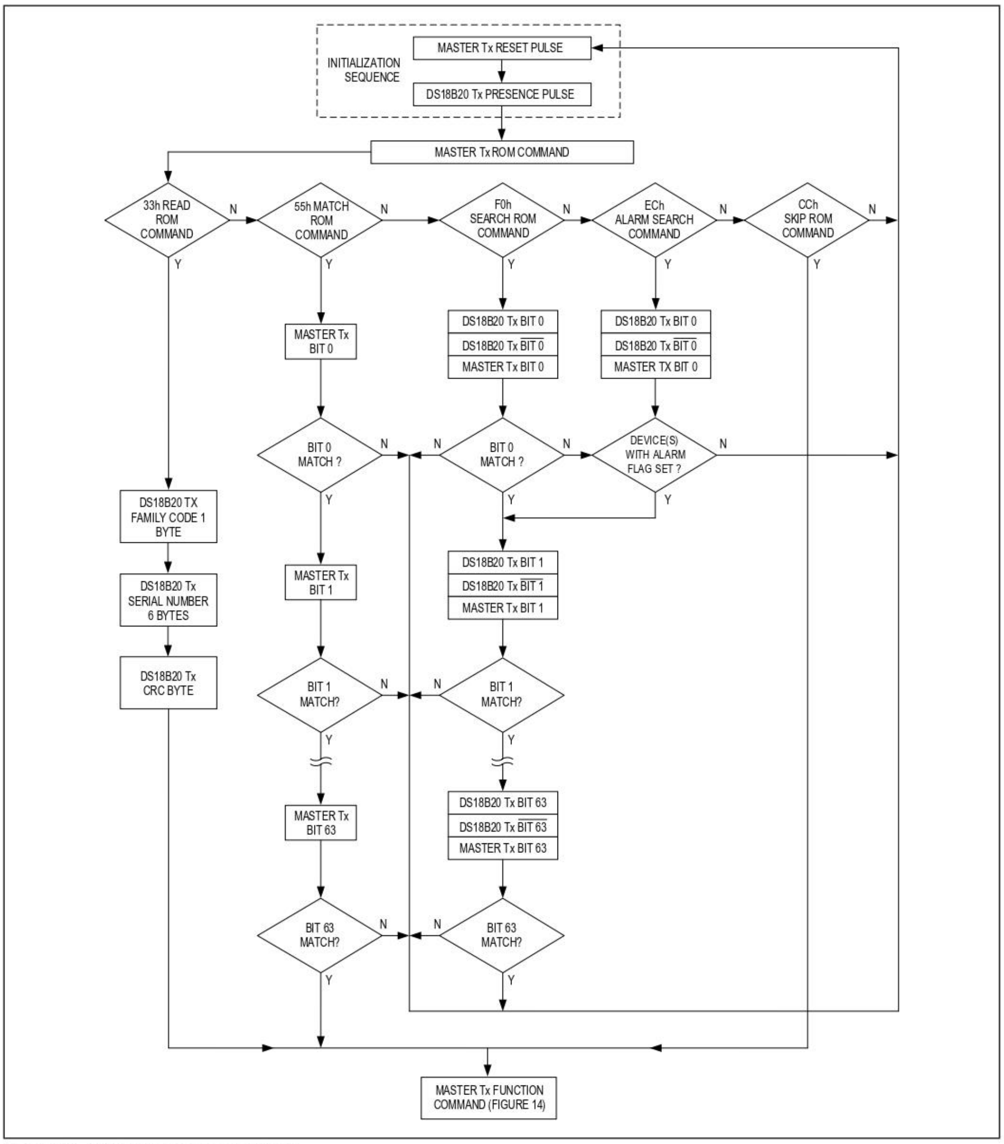


Figure 13. ROM Commands Flowchart

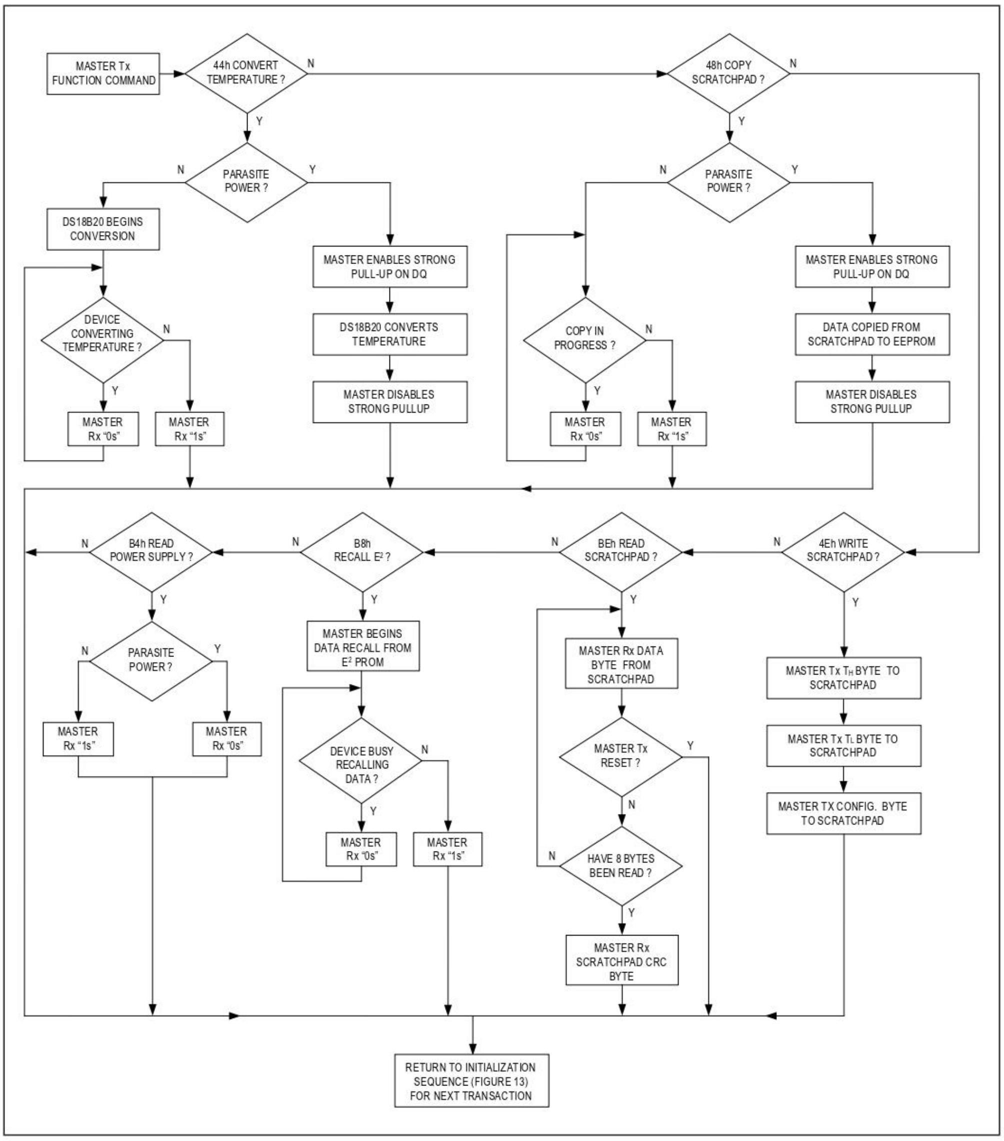


Figure 14. DS18B20 Function Commands Flowchart

1-Wire Signaling

The DS18B20 uses a strict 1-Wire communication protocol to ensure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. The bus master initiates all these signals, with the exception of the presence pulse.

Initialization Procedure—Reset And Presence Pulses

All communication with the DS18B20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18B20. This is illustrated in Figure 15. When the DS18B20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (T_X) the reset pulse by pulling the 1-Wire bus low for a minimum of 480µs. The bus master then releases the bus and goes into receive mode (R_X). When the bus is released, the 5kΩ pullup resistor pulls the 1-Wire bus high. When the DS18B20 detects this rising edge, it waits 15µs to 60µs and then transmits a presence pulse by pulling the 1-Wire bus low for 60µs to 240µs.

Read/Write Time Slots

The bus master writes data to the DS18B20 during write time slots and reads data from the DS18B20 during read time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

Write Time Slots

There are two types of write time slots: “Write 1” time slots and “Write 0” time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18B20 and a Write 0 time slot to write a logic 0 to the DS18B20. All write time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (see Figure 14).

To generate a Write 1 time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within 15µs. When the bus is released, the 5kΩ pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-Wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least 60µs).

The DS18B20 samples the 1-Wire bus during a window that lasts from 15µs to 60µs after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18B20. If the line is low, a 0 is written to the DS18B20.

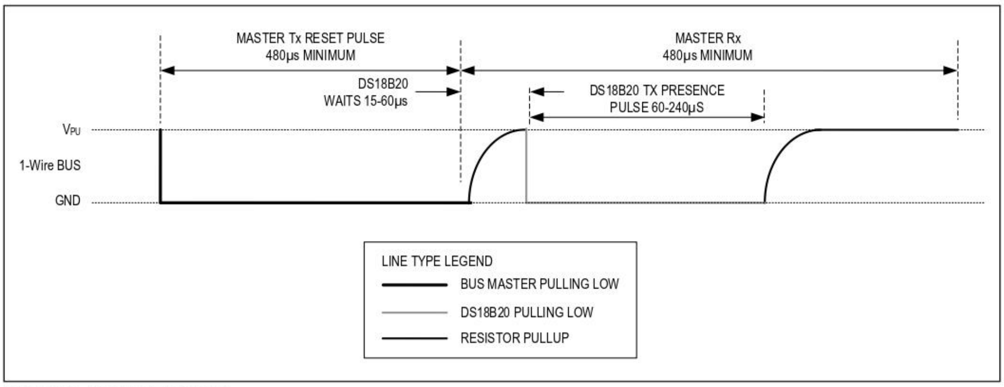


Figure 15. Initialization Timing

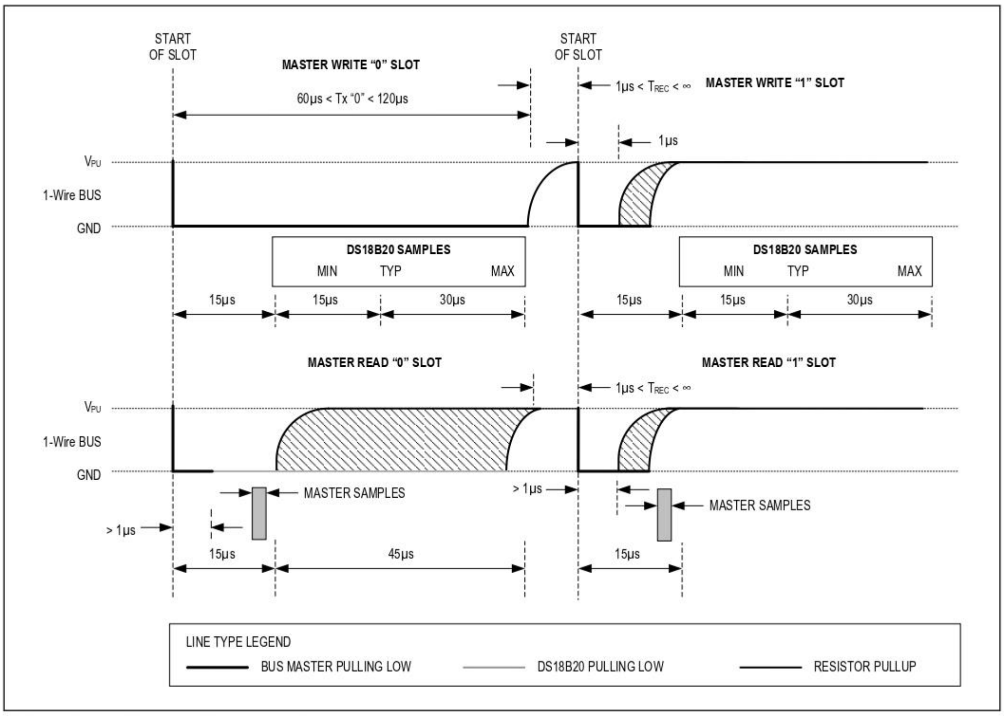


Figure 16. Read/Write Time Slot Timing Diagram

Read Time Slots

The DS18B20 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18B20 can provide the requested data. In addition, the master can generate read time slots after issuing Convert T [44h] or Recall E^2 [B8h] commands to find out the status of the operation as explained in the *DS18B20 Function Commands* section.

All read time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (see Figure 16). After the master initiates the read time slot, the DS18B20 will begin transmitting a 1 or 0 on bus. The DS18B20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18B20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resister. Output data from the DS18B20 is valid for 15µs after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within 15µs from the start of the slot.

Figure 17 illustrates that the sum of T_{INIT}, T_{RC}, and T_{SAMPLE} must be less than 15µs for a read time slot. Figure 18 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as short as possible and by locating the master sample time during read time slots towards the end of the 15µs period.

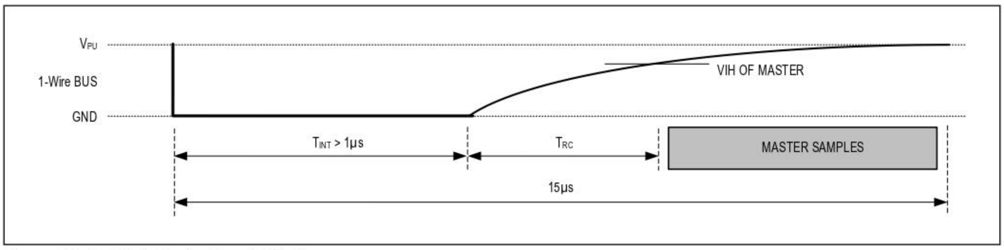


Figure 17. Detailed Master Read 1 Timing

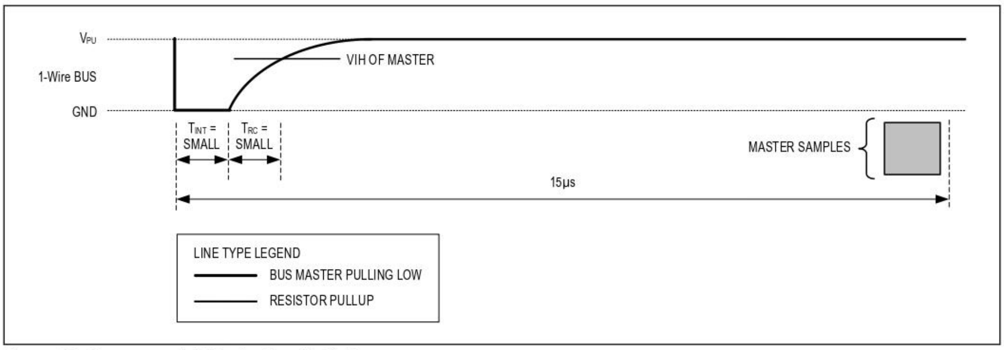


Figure 18. Recommended Master Read 1 Timing

Related Application Notes

The following application notes can be applied to the DS18B20 and are available at www.maximintegrated.com.

Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products

Application Note 122: Using Dallas' 1-Wire ICs in 1-Cell Li-Ion Battery Packs with Low-Side N-Channel Safety FETs Master

Application Note 126: 1-Wire Communication Through Software

Application Note 162: Interfacing the DS18x20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment

Application Note 208: Curve Fitting the Error of a Bandgap-Based Digital Temperature Sensor

Application Note 2420: 1-Wire Communication with a Microchip PICmicro Microcontroller

Application Note 3754: Single-Wire Serial Bus Carries Isolated Power and Data

Sample 1-Wire subroutines that can be used in conjunction with *Application Note 74: Reading and Writing iButtons via Serial Interfaces* can be downloaded from the Maxim website.

DS18B20 Operation Example 1

In this example there are multiple DS18B20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18B20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion (t_{CONV}).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

DS18B20 Operation Example 2

In this example there is only one DS18B20 on the bus and it is using parasite power. The master writes to the TH, TL, and configuration registers in the DS18B20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	4Eh	Master issues Write Scratchpad command.
Tx	3 data bytes	Master sends three data bytes to scratchpad (T_H, T_L, and config).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	48h	Master issues Copy Scratchpad command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 FSOP	18B20
DS18B20U+	-55°C to +125°C	8 FSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 FSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 FSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

+Denotes a lead-free package. A "+" will appear on the top mark of lead-free packages.

T&R = Tape and reel.

**TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.*

Revision History

REVISION DATE	DESCRIPTION	PAGES CHANGED
030107	In the Absolute Maximum Ratings section, removed the reflow oven temperature value of +220°C. Reference to JEDEC specification for reflow remains.	19
101207	In the *Operation—Alarm Signaling* section, added "or equal to" in the description for a TH alarm condition	5
	In the *Memory* section, removed incorrect text describing memory.	7
	In the *Configuration Register* section, removed incorrect text describing configuration register.	8
042208	In the *Ordering Information* table, added TO-92 straight-lead packages and included a note that the TO-92 package in tape and reel can be ordered with either formed or straight leads.	2
1/15	Updated *Benefits and Features* section	1
09/18	Updated *DC Electrical Characteristics* table	2

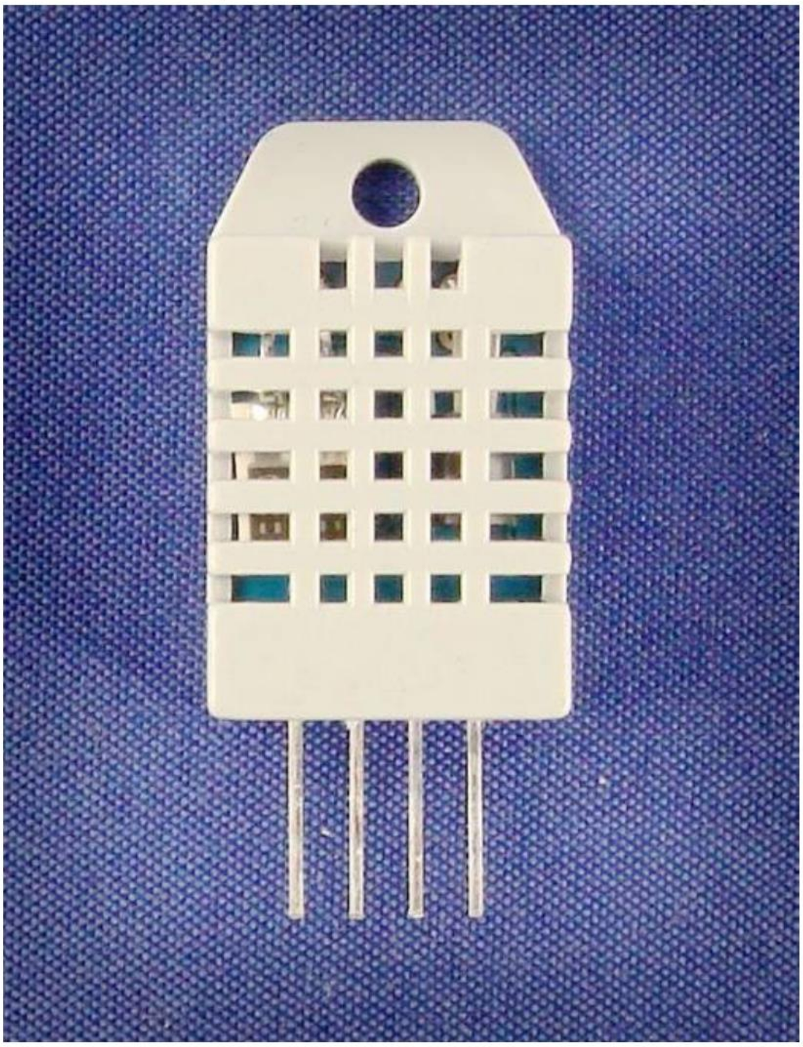
Annex VI: *Datasheet* del DHT 22

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

Digital-output relative humidity & temperature sensor/module

DHT22 (DHT22 also named as AM2302)



Capacitive-type humidity and temperature module/sensor

1

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

1. Feature & Application:

* Full range temperature compensated * Relative humidity and temperature measurement
* Calibrated digital signal *Outstanding long-term stability *Extra components not needed
* Long transmission distance * Low power consumption *4 pins packaged and fully interchangeable

2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability.Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH
Long-term Stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

4. Dimensions: (unit----mm)

1) Small size dimensions: (unit----mm)

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

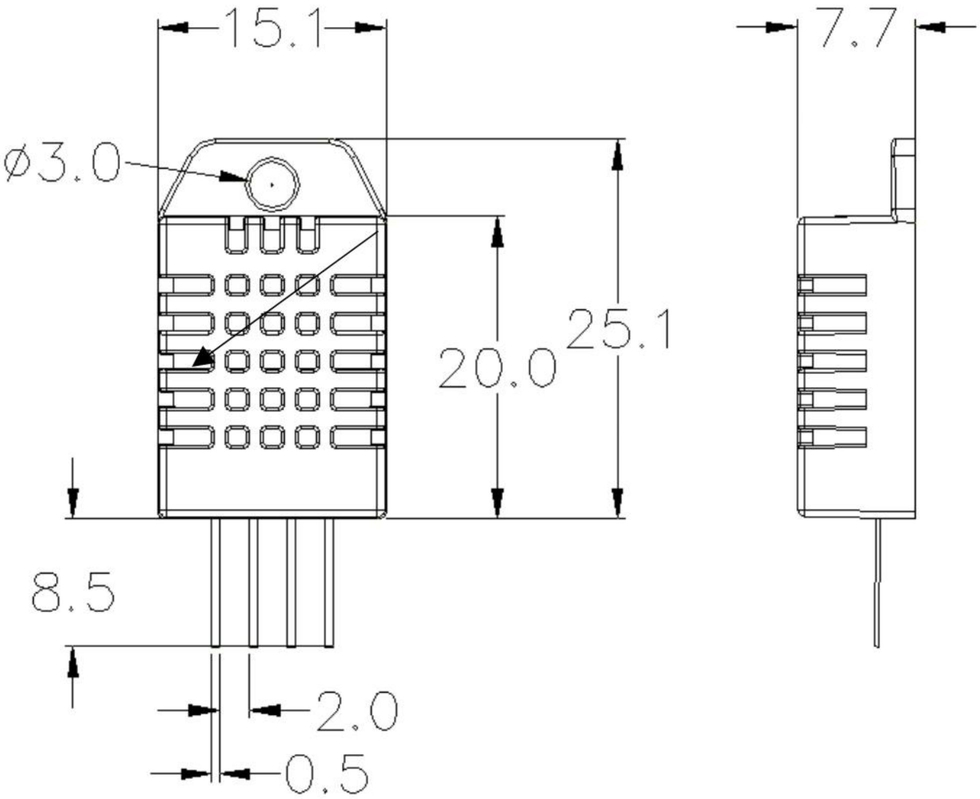
3

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD----power supply
2	DATA--signal
3	NULL
4	GND

4

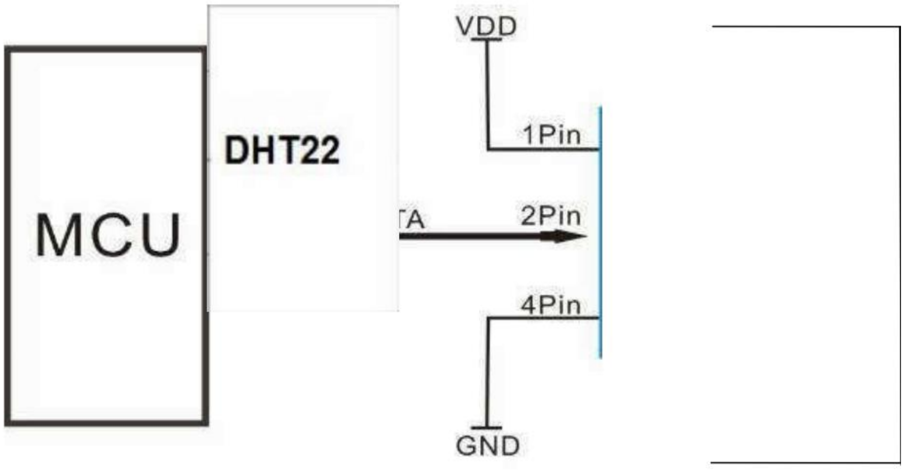
Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

5. Electrical connection diagram:



3Pin---NC, AM2302 is another name for DHT22

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

Single-bus data is used for communication between MCU and DHT22, it costs 5mS for single time communication.

Data is comprised of integral and decimal part, the following is the formula for data.

DHT22 send out higher data bit firstly!
DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum
If the data transmission is right, check-sum should be the last 8 bit of "8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data".

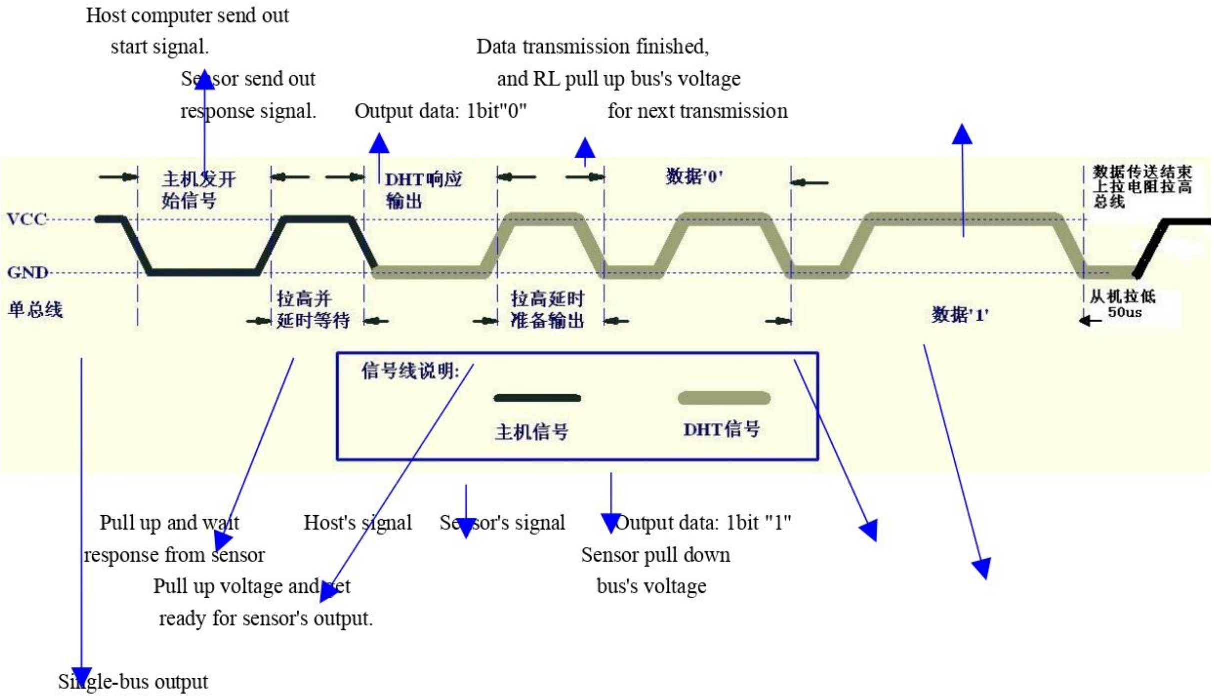
When MCU send start signal, DHT22 change from low-power-consumption-mode to running-mode. When MCU finishs sending the start signal, DHT22 will send response signal of 40-bit data that reflect the relative humidity

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

and temperature information to MCU. Without start signal from MCU, DHT22 will not give response signal to MCU. One start signal for one time's response data that reflect the relative humidity and temperature information from DHT22. DHT22 will change to low-power-consumption-mode when data collecting finish if it don't receive start signal from MCU again.

1) Check bellow picture for overall communication process:



2) Step 1: MCU send out start signal to DHT22

Data-bus's free status is high voltage level. When communication between MCU and DHT22 begin, program of MCU will transform data-bus's voltage level from high to low level and this process must beyond at least 1ms to ensure DHT22 could detect MCU's signal, then MCU will wait 20-40us for DHT22's response.

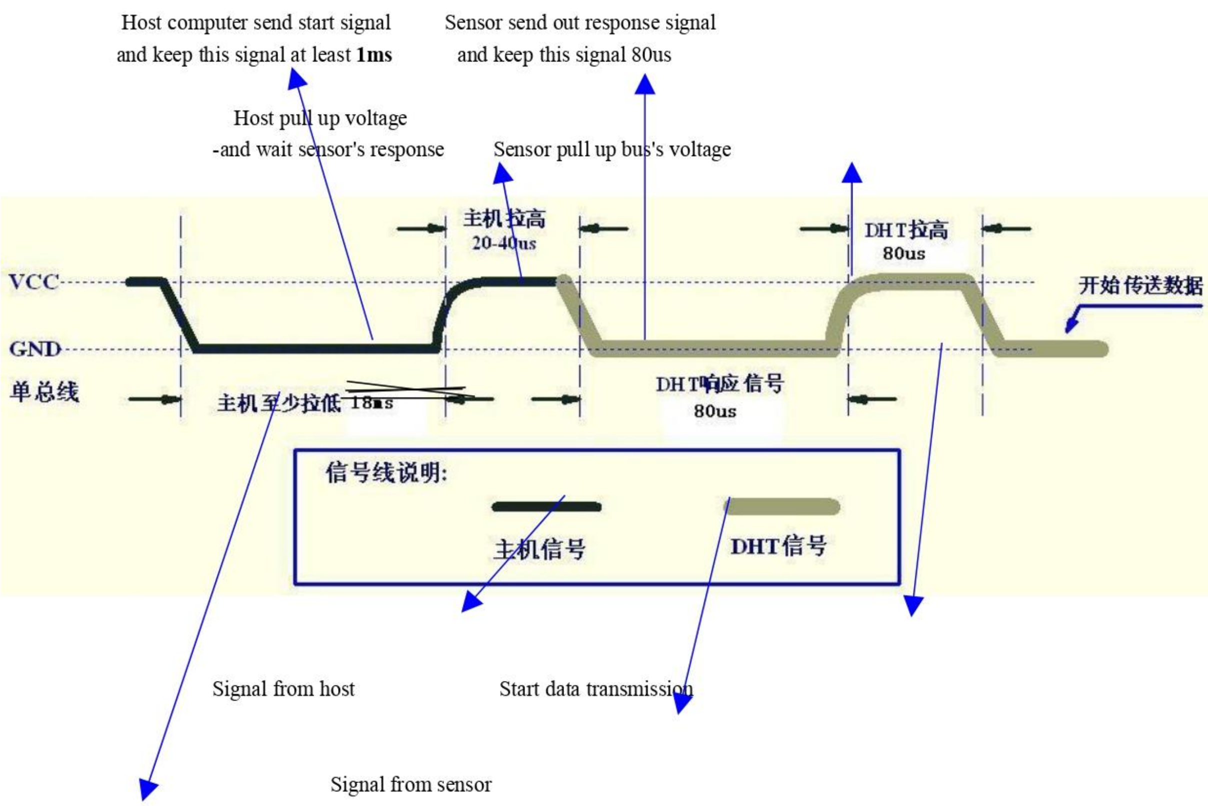
Check bellow picture for step 1:

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



Single-bus signal

Step 2: DHT22 send response signal to MCU

When DHT22 detect the start signal, DHT22 will send out low-voltage-level signal and this signal last 80us as response signal, then program of DHT22 transform data-bus's voltage level from low to high level and last 80us for DHT22's preparation to send data.

Check bellow picture for step 2:

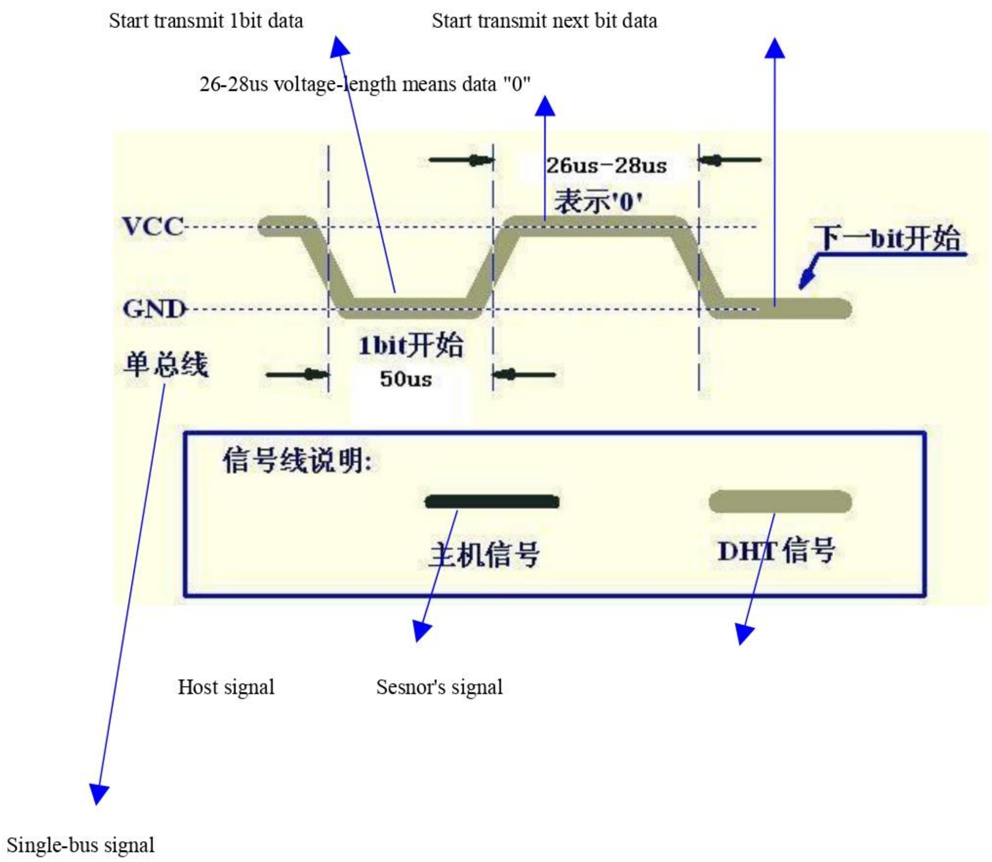
7

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



Step 3: DHT22 send data to MCU

When DHT22 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0".

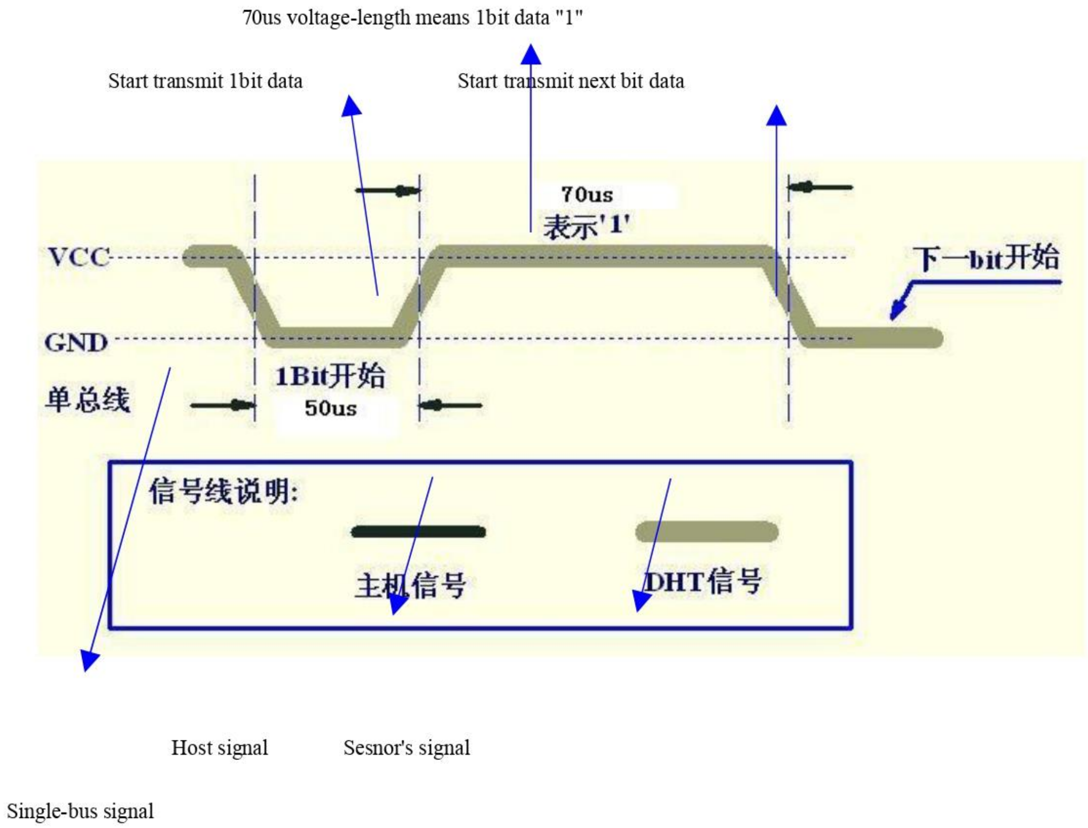
Check bellow picture for step 3:

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



If signal from DHT22 is always high-voltage-level, it means DHT22 is not working properly, please check the electrical connection status.

7. Electrical Characteristics:

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

*Collecting period should be : >2 second.

9

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The DHT22 sensor can recover after working in non-normal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere DHT22's sensitive-elements and debase DHT22's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the DHT22 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the DHT22 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

DHT22 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase DHT22's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius.
* Avoid using the sensor under dew condition.
* Don't use this product in safety or emergency stop devices or any other occasion that failure of DHT22 may cause personal injury.

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Annex VII: Declaració d'honor

I declare that,

the work in this Master Thesis is completely my own work,

no part of this Master Thesis is taken from other people's work without giving them credit,

all references have been clearly cited,

I'm authorised to make use of the research group related information I'm providing in this document.

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by *The Universitat Politècnica de Catalunya - BarcelonaTECH*.

Guillem Llop Alonso		20 / 06 / 2019
Student Name	Signature	Date

Title of the Thesis:

Estudi d'un sistema de supervisió remota basat en sensors intel·ligents